Nintendo®

ベーシックで広がる、ファミリーコンピュータ™の世界。

ファミリーベーシック™で遊ぼう！



㈱O＆Kマネージメントサービス
鹿島 博

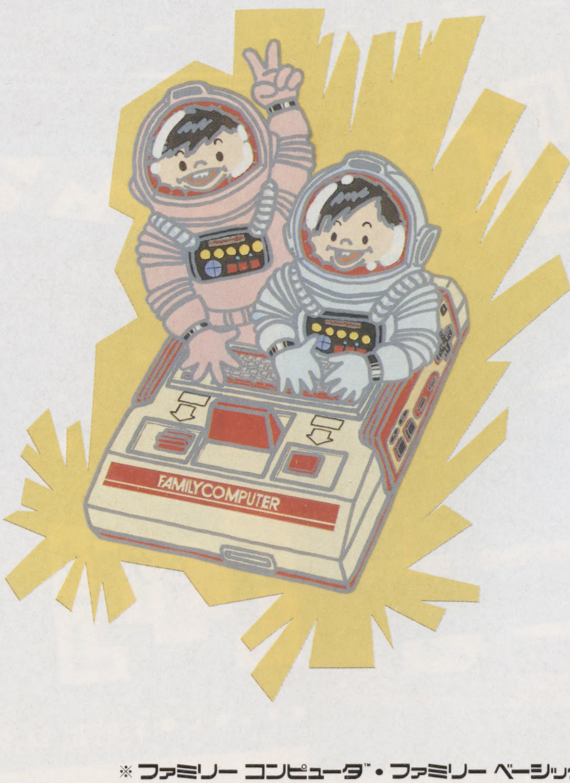
マイクロデザイン

Nintendo®

——ベーシックで広がる、 ファミリー コンピュータ™の世界。——

ファミリー ベーシック™で遊ぼう!

㈱O＆Kマネージメントサービス
鹿島　博

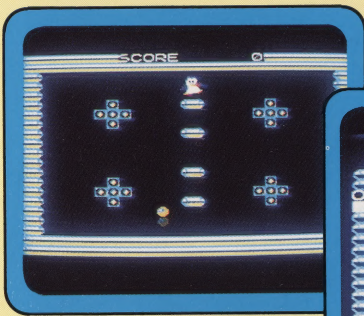


※ ファミリー コンピュータ™・ファミリー ベーシック™ は、任天堂の商標です。

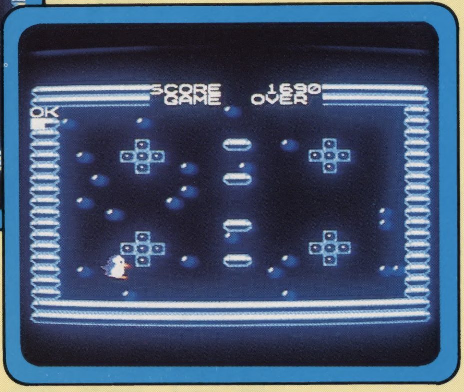
こんなゲームあんなゲーム

ファミリーベーシックだからカンタン

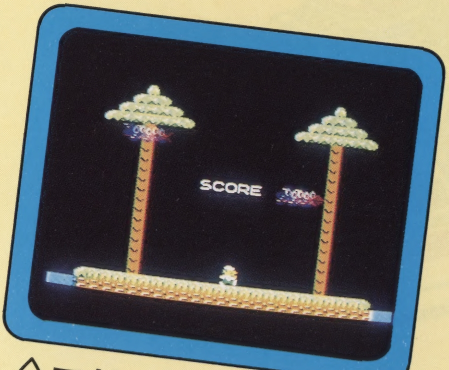
♡ペンペン大追跡(だいついせき)



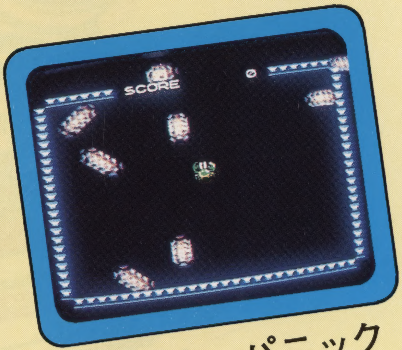
▲オープニング画面



終了画面▶



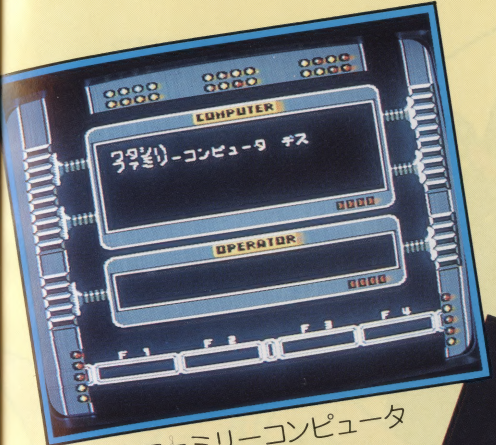
♤マリオランド



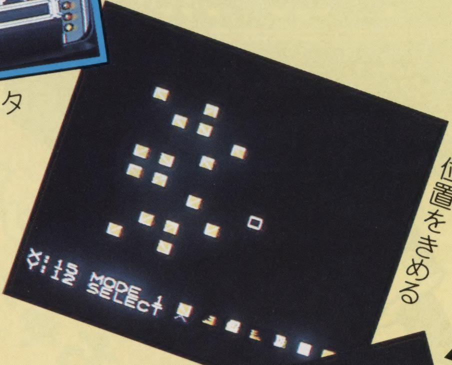
♧スピナーパニック

らくらくベーシックで君のイメージをふくらまそう

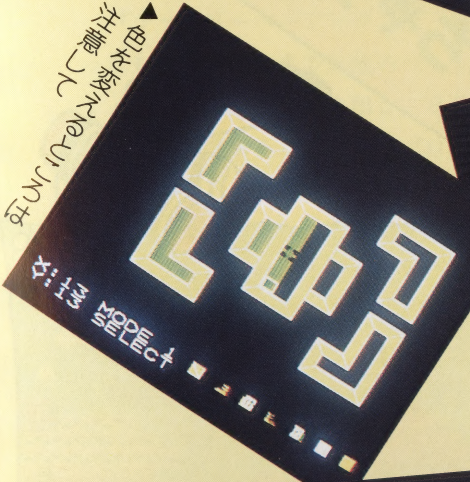
BG　グラフィックモードで
ゲームの背景を作る過程を目
で見てたしかめよう！
これからご紹介する画面は，
「チェンジ・チェンジ」です。



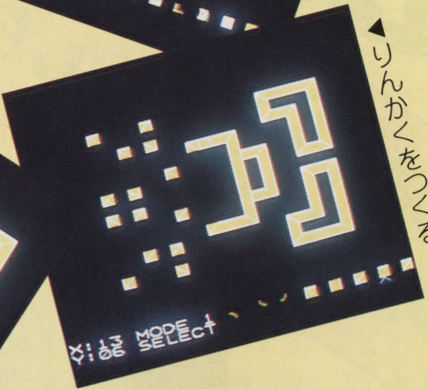
▲まず，ファミリーコンピュータにBGグラフィックモードを出してもらう



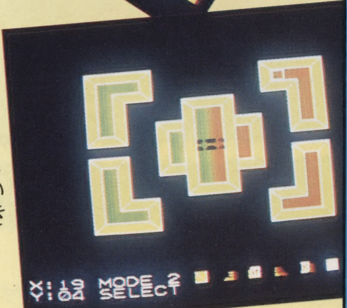
▲位置をきめる



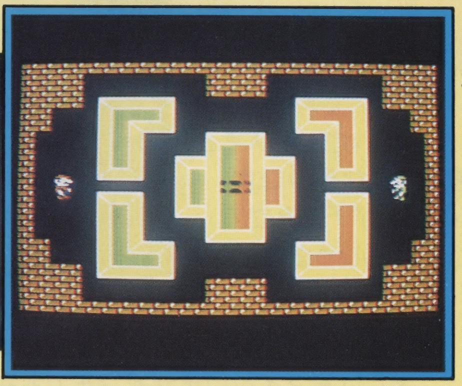
▲色を変えるところは注意して



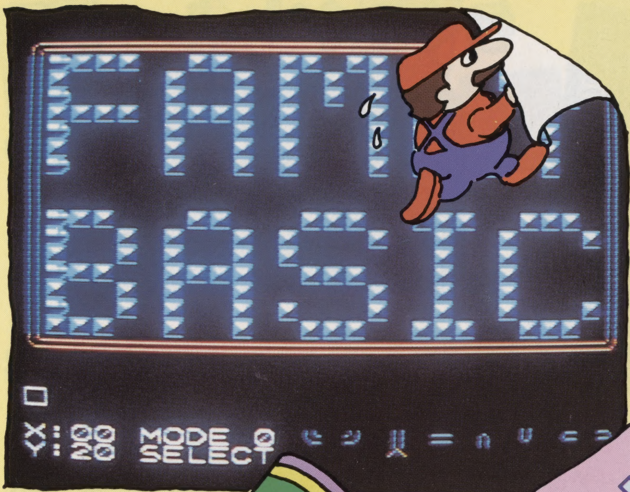
▲りんかくをつくる



▲あとひといき



▲まわりをかためたら完成！

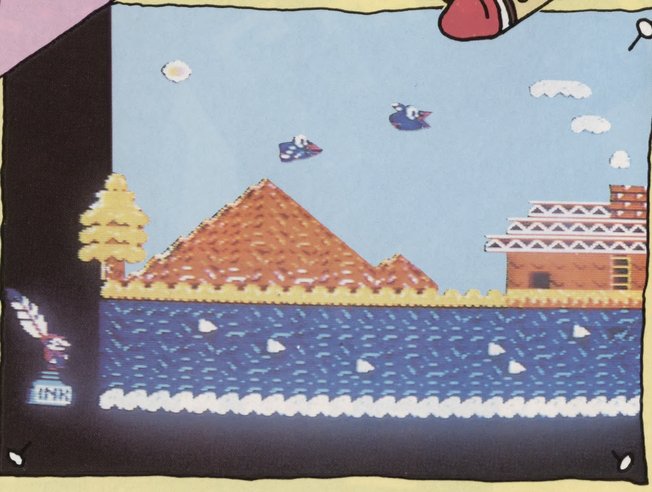


せっかくのロゴマークだから、色をつけて、形も整えてあげよう。



グラフィックだけでも楽しめる!!
＝組み立てお絵描きしちゃうのだ＝





インベーダーだって使い方しだいで波や雲になるのです。このへんが「組み立て式」のおもしろさだネ。

＊本書の内容に対するご意見・ご質問は㈱O＆Kマネージメントサービスまたは㈱マイクロデザインまでお寄せください。

はじめに

この本は，任天堂の『ファミリーコンピュータ』用『ファミリーベーシック』を使って，思いっきり遊びたい人のために書いたものです。

『ファミリーコンピュータ』は，すでに150万台を突破するほどの売れ行きを見せているベストセラー機です。

そして『ファミリーベーシック』は，その『ファミリーコンピュータ』に接続すると，自分でコンピュータのプログラムが組めるという，画期的な製品なのです。専用のゲームカートリッジで遊ぶだけではなく，本格的なパソコンと同じように，自分でプログラムを作って，いろいろな用途に利用することもできます。

『ファミリーベーシック』は，数あるベーシックの中でも，ちょっと変わったものです。今までに他のベーシックを扱ったことのある人ならわかるでしょうが，画面に人間を出し，それをいろいろな方向に歩かせるためには，めんどうな準備をしてから，プログラムを組まなければなりませんでした。ところが『ファミリーベーシック』では，たとえば，マリオを画面に出して右向きに歩かせるという動作が，たった4つの命令でできてしまうのです。ですから初めて扱う人でも，すぐに楽しむことができると思います。

これからの社会では，コンピュータのことをまったく知らない，というわけにはいかないでしょう。あまりカタイことをいわなければ，遊びながらでも知識を身につけることはできるはずであり，そんな使い方に『ファミリーベーシック』は最適です。

この本では，むずかしい理屈はちょっと横に置いといて，とにかく『ファミリーベーシック』にさわって，動かして，体で覚えてしまえる内容にしました。巻末には，4種類のゲームプログラムも載せてあります。

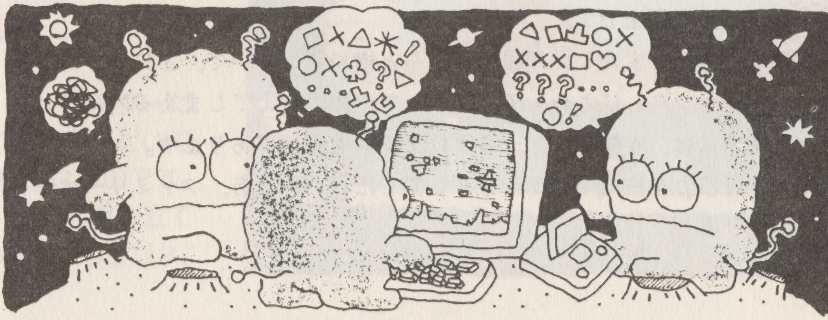
では，肩の力を抜いてリラックスしてください。さあ，ファミリーベーシックで遊びましょう。

もくじ

第1章 まずは基礎編

第2章 テクニック編

第3章 BG グラフィック編



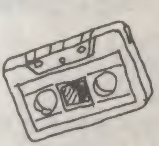
表紙・本文イラスト 佐々木正樹

もくじ

第4章　ハイ・テク編

第5章　効果テキメン編

第6章　ゲームライブラリー



★BASIC・V2.0・テープサービスのご案内

本書に掲載したプログラムのテープサービスを行なっております。キー入力の手間を省いて，プログラムの勉強をスピーディに進めるためにお役立てください。ご注文は(株)マイクロハウス（宛先は巻末奥付参照）宛，1800円(送料共)を現金書留にてお送りください。（注：BASIC・V1.0のテープサービスは行なっておりません。）

注意　この本で掲載したプログラムは，NS−HUBASIC V2.0用です。BASICモードに入ったときにV1.0と表示される製品では動きません。

第1章 まずは基礎編



これから，マリオやペンペンなどのゲームキャラクタを画面に出すことから始めて，キャラクタが物を取ったり，何かにぶつかったなどの判定をすることまでのやり方を，順を追ってやっていきます。それぞれのテクニックが，ゲームプログラムを組むときの部品になるようなものと考えてください。

1. プログラムを打ち込むときの注意

さて，この本にはテクニックを覚えるために，いろいろなプログラムリストが出てきます。こうしたプログラムを打ち込むために，次のことをぜひ覚えておきましょう。

●きれいさっぱり、新しく

新しくプログラムを打ち込むときには，打ち込みを始める前に N E W RETURN と打ってください。これは,「これから新しくプログラムを覚えるから，今までのプログラムは全部忘れなさい」という命令です。では，

```
100 PRINT A
```

というプログラムを打ち込んでみましょう。たった1行でも，100という行番号が付いていますから，れっきとしたプログラムなのです。

1 Ø Ø SPACE P R I N T SPACE A

と、順番に打ち込んでいってください。そして、SHIFT キーを押さえたまま CLR HOME キーを押して，一度画面を消してしまいましょう。

●プログラムを覚えさせる

次に

[L][I][S][T] [RETURN]

と、打ってみましょう。これは，ファミリーコンピュータが覚えているプログラムを、画面に表示させる命令です。

さて、画面には何が表示されましたか？　おやっ？　「ＯＫ」の文字のほかは，何も出ませんね。ということは，ファミリーコンピュータは，さっき打ったプログラムを覚えてくれていないのです。どうしてでしょうか？

その秘密(ひみつ)は，[RETURN] キーにあるのです。では今度は，

[1][Ø][Ø] [SPACE] [P][R][I][N][T] [SPACE] [A] [RETURN]

と、打ってみてください。そして、[SHIFT] ＋ [CLR HOME] キーを押し，画面を消してから，[L][I][S][T] [RETURN] と打つと……，どうですか？　ちゃんとプログラムが出てきましたね。

つまり，このときの [RETURN] キーは，“打ち終わった行を覚える”という働きを持っているので，プログラムを１行打ち終えたら，必ず [RETURN] キーを押さなくてはならないのです。

●プログラムを直したいとき

次は，ファミリーコンピュータが覚えているプログラムの一部を直したいときの方法を練習しましょう。

今画面には，さっき呼(よ)びだしたプログラムが出ていますね。出ていない人は，[L][I][S][T] [RETURN] で出してください。さて，今ファミリーコンピュータは，「1ØØ　PRINT　A」というプログラムを覚えているのですが，これを,「1ØØ　PRINT　B」に直したいとします。こういうときは矢印キーを使って，ピコピコ点滅(てんめつ)しているもの（カーソルと呼びます）をＡの上に来るまで動かしてください。そして [B] [RETURN] と打ちます。これでＡがＢに変わりましたね。このときに押す [RETURN] キーは，「この行はこう変わったから，このとおり覚えな



さい」という働きをします。

では，確かめてみましょう。画面を消して([SHIFT] + [CLR HOME]),リストを出して（[L][I][S][T][RETURN]）みてください。「100　PRINT　B」と出ましたか？　出ていればオーケーです。

次に,「100　PRINT　B」を「100　PRINT　AB」に直してみます。矢印キーでカーソルを「B」のところへ動かし，[INS]　[A]　[RETURN] と打ってください。[INS] キーは，1度押すとカーソルのある位置を，1字あける働きをします。打ち終わったら，また画面を消しリストを出して,「100　PRINT　AB」と直っていることを確かめておきましょう。

最後にもう1つ。「100　PRINT　AB」を「100　PRINT　B」に直す練習をしておきましょう。矢印キーを使って，カーソルを「B」のところに動かし，[DEL]　[RETURN] と打ってください。[DEL] キーは，カーソルの左側を1文字消して後をつめる働きを持っていますから,「AB」の「A」が消え,「B」だけになりましたね。直したあとは，また新しくリストを出して，確認(かくにん)しておいてください。ちゃんと,「100　PRINT　B」となっていますね。

さて，今までやってきたことは，プログラムを打ち込むために，ぜひ，覚えておいてほしいことですから，めんどうがらずに練習しておいてください。



2. 内蔵(ないぞう)キャラクタを画面に出してみよう

●ペンペン登場(とうじょう)

いよいよ画面に何かを出して動かしてみることにしましょう。まず，何も考えずに次のプログラムを打ち込んでください。

```
100 CLS                                      ←画面をきれいにする
110 SPRITE ON                                ←内蔵キャラクタを表示する準備をする
120 DEF MOVE(Ø)=SPRITE(4, 3, 2               ←ペンペンをどう動かすかを決める
, 128, Ø, Ø)
130 POSITION Ø, 120, 100                     ←ペンペンの動きだす最初の位置を決める
140 MOVE Ø                                   ←ペンペンを動かす
```

リスト 1

打ち込み終わったら，R U N RETURN と押してみてください。どうなりましたか？ 画面のまんなかあたりにペンペンが現れて，右に向かって歩き始めましたね。そして，画面の右端で消えたと思うと，今度は左端に登場してきて，はじめに現れた場所で止まります。これが，内蔵キャラクタを画面に出して動かす，一番簡単なやり方です。ペンペンの色がおかしいですが，色のことは後でやりますから，それまでガマンしていてください。

さて，今のプログラムの120行に，「DEF MOVE……」 という長い命令が出てきましたが，この命令はたいへん便利な命令なのです。詳しく説明すると，次のようなことになっています。

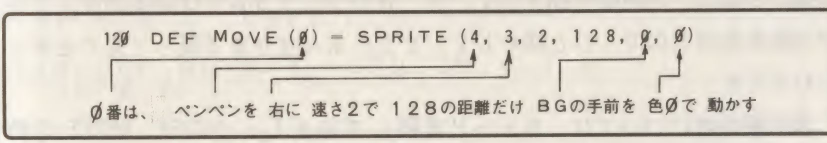


図 1

1つの命令で，ずいぶんいろんなことを決めていますね。

●マリオに交代

それでは，上の120行の「4」(ペンペンを)のところを，ちょっと変えてみましょう。画面にリストを出し，「4」を「Ø」に直してもう一度，R U N RETURN としてみてください。すると今度は，ペンペンではなくマリオが出てきて，前と同じように動きましたね。つまり，「ペンペンを」（4）が，「マリオを」(Ø）に変わったわけです。このように，「何々を」のところに，Ø～15までの数字をどれか1つ入れると，それぞれの数字に対応し

0	マリオ
1	レディ
2	ファイターフライ
3	アキレス
4	ペンペン
5	ファイアーボール
6	車
7	スピナー
8	スターキラー
9	スターシップ
10	爆発
11	ニタニタ
12	レーザー
13	シェルクリーパー
14	サイドステッパー
15	ニットピッカー

図 2

た内蔵キャラクタが出てきて，動かすことができるのです。

さあ，いろいろな数字に変えてキャラクタを動かしてみましょう。

●方向や速さを変える

「何々を」と同じように，「右に」や「速さ２で」のところも数字を変えてやると，動く方向や速さが変わります。でも，どんな数字でもよいというわけではなく，図３のように，それぞれ決まりがあります。

方　　向	速　　度	全移動量	表示優先度	配色番号
8 ＼ 1 ／ 2 7 ― 方向 ― 3 6 ／ 5 ＼ 4	速い←　→遅い 1 ～ 255	1 ～ 255	Ø：バックグラウンドより前に表示 1：バックグラウンドの後に表示	Ø～3
Øだと止まったパターンになります	Øだとほとんど動きません	Øだと表示されません		

図　３

全移動量というのがありますが，この数字を128にすると，ちょうど画面の端から端までの距離を動きます。さっきのプログラムでは画面のまんなかから動き始めたので，ひと回りして，また，まんなかまで戻って来て止まったわけです。

表示優先度については，ちょっと実験してみましょう(DEF MOVE の数字をいろいろ変えてしまった人は，もとに戻しておいてください)。

まず，120行の「方向」の数字を「Ø」にしてＲＵＮしてください。ペンペンが画面のまんなかへんに，こちらを向いて出ていますね。では，ペンペンのいる位置に重ねて，何か文字を打ってみてください。どうなりましたか?

ペンペンのところだけ，文字が見えませんね。

それでは，またリストを出して，今度は「表示優先度」を「１」に変えて，もう一度ＲＵＮしてください。そして同じように，ペンペンのいる場所に何か文字を打ってみましょう。今度は，文字がちゃんと見えますね。つまり，「表示優先度」を「Ø」にすると，バックの絵や文字のこちら側にペンペンが見え，「１」にすると，絵や文字のむこう側にペンペンが見えるということなのです。この文字の書かれている面のことをバックグラウンド面と呼び，BG GRAPHICモードで描いた絵は，この面に描かれ,「表示優先度」を「Ø」や「１」にすることによって，画面に遠近感を出すこともできるわけです。

●内蔵キャラクタの色を変える

さて，さっき約束した色について，ここで説明しておきましょう。今ペンペンは，青・水色・白の3色になっていますね。何だか寒そうですから，色を変えてあげることにしましょう。さっきのプログラムの100行，

```
100 CLS
```

を，

```
100 CLS : CGSET ,2
```

のように変えて，もう一度ＲＵＮしてみてください。

どうですか？　少し赤みが付いて，ペンペンらしい色になりましたね。このように，ペンペンの色は，100行のCGSET命令と，120行のDEF MOVE命令の中の配色番号の2つによって決まっているのです。ちょっと図４を見てください。

```
100 CLS : CGSET ,2
120 DEF MOVE(0)=SPRITE(4,3,2,128,0,0)
```

	CGSET ,0			CGSET ,1			CGSET ,2		
配色番号 0	54	22	2	48	22	1	48	38	18
配色番号 1	39	48	25	16	0	1	48	21	18
配色番号 2	53	37	23	48	41	9	48	18	22
配色番号 3	48	39	22	48	22	7	48	38	25

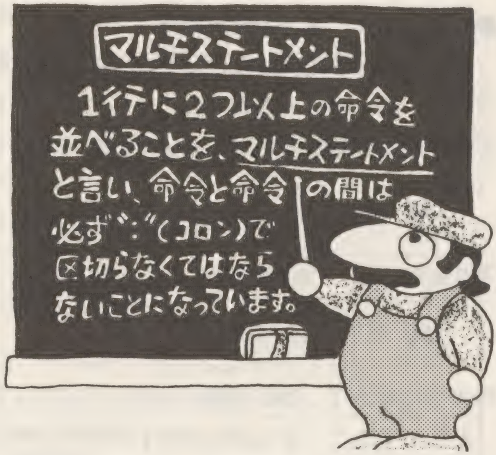
図　4

まず，100行の「CGSET, 2」で，12組の組み合わせのうち4組が選ばれ，次に120行の「DEF MOVE」の配色番号「0」で，その4組のうち「48　38　18」という組み合わせがさらに選ばれ，ペンペンの色が決められたわけです。この「48」,「38」,「18」という数字は，それぞれの色を番号で示したもので，「48」は体の白,「38」はくちばしのオレンジ色,「18」は背中の青のことです。

では，またちょっと色を変えてみましょう。「CGSET, 0」，配色番号「2」

にして，RUNしてみてください。なんとも変な色のペンペンになってしまいました。失礼…。

こうして，CGSET命令と配色番号の指定を組み合わせることによって，色が選べることがわかりましたね。



●キャラクタの位置をきめて動かす

さて，次にPOSITIONとMOVEについて少し説明しておきましょう。POSITIONは，DEF MOVEで決めた，キャラクタの動き始める画面上の位置を指示する命令です。リスト1の130行でいえば，「12Ø」が横方向「1ØØ」が縦方向の位置を表しています。では，縦方向の位置を「1ØØ」から「2ØØ」に変えてRUNしてみてください。ペンペンが，画面の下の方に出てきましたね。このように，縦方向では上へ行くほど小さく，下へ行くほど大きな数字になります（Ø～255)。また横方向では左へ行くほど小さく，右へ行くほど大きな数字になります（Ø～255)。

MOVEは，DEF MOVE命令とPOSITION命令によって決めたいろいろなことを，実際に行わせる命令です。たとえば140行をなくしてしまって（1 4 Ø RETURN で消えます）RUNさせてみてください。画面には何も表れてくれません。つまり，DEF MOVE, POSITION, MOVEは，3つひと組で使うのだということがわかります。またこのとき，

```
12Ø DEF MOVE (Ø) = ・・・・・・・・・
                  |
13Ø POSITION Ø, 12Ø, 1ØØ
              ┌────┘
14Ø MOVE Ø
         |
         └─── 数字が同じ" Ø"です
```

図 5

というように，３つの命令に出てくる数字を合わせておかなければなりません。この数字は，Ø～７までの８通り使えますが，たとえば，DEF MOVE（3）＝……としたときは，後の２つも同じ「3」にするということです。

では，１つサンプルプログラムを紹介しておきましょう。このプログラムは，内蔵キャラクタを次々に登場させるもので，そのままにしておくといつまでも動き続けますから，見あきたら STOP キーで止めてください。

なお，画面に残ったキャラクタは，S P R I T E O F F RETURN と打つと消えます。

```
100 CLS:CGSET 1,1:LOCATE 9,8
:PRINT"パターン ナンバー"
110 SPRITE ON
120 FOR J=Ø TO 15
130 LOCATE 12,1Ø:PRINT J;"  "

140 FOR I=Ø TO 7
150 DEF MOVE(I)=SPRITE(J,I+1
,1,255,Ø,Ø)
160 POSITION I,12Ø,12Ø
170 MOVE I
180 NEXT
190 PAUSE 33Ø
200 NEXT
210 GOTO 12Ø
```

リスト 2

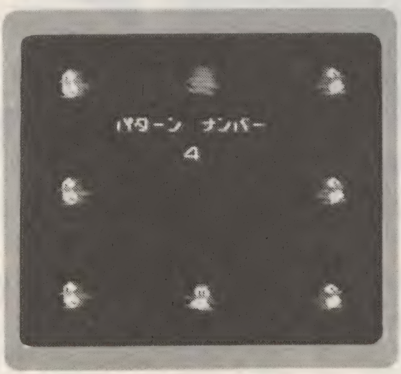


写真 1

3. プログラムを SAVE,LOAD する方法

●SAVE命令で記録する

さて，今ファミリーコンピュータは，前項で打ち込んだプログラムを覚えています。ここで次のプログラムを打ち込むために，ＮＥＷ命令でこのプログラムを忘れさせてしまうのは，もったいないですから，テープに記録（ＳＡＶＥ）しておきましょう。

まず，テープレコーダをつないでください。キーボードのうしろにある「WRITE」がテープレコーダの録音（ＲＥＣまたはＭＩＣ）端子に，「ＲＥＡＤ」が再生（ＭＯＮＩＴＯＲまたはEAR）端子にちゃんとつながっているかを，よく確かめましょう。

ではまず，記録するプログラムに付ける名前を考えましょう。あまり長いとめんどうですから，「ＰRO」にしましょう。では次のように打ってください。

S A V E “ P R O ”

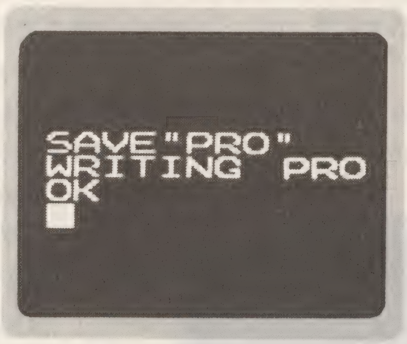


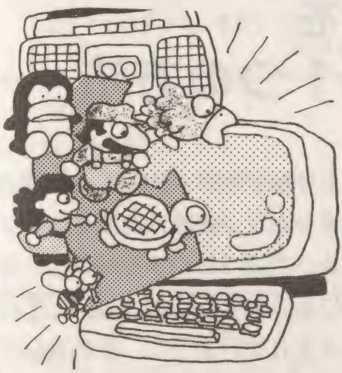
写真　2



SAVE がテープレコーダに記録する命令で，PRO が名前です。打ち終わったら，テープレコーダの録音ボタンを押してから，RETURN キーを押してください。すると画面に，「WRITING PRO」とメッセージが出て，しばらくして「ＯＫ」と表示されます。これでテープへの記録が終わりましたから，テープレコーダを止めてください。

●呼び出しはLOAD命令

テープに記録したプログラムを，ファミリーコンピュータに呼び出すときは，LOAD命令を使います。今，SAVEしたプログラムを呼び出してみましょう。まず N E W RETURN で，ファミリーコンピュータにプログラムを忘れさせてしまってください。そして，テープを巻き戻して，L O A D RETURN と打ち込んで，テープレコーダの再生ボタンを押します。このとき，レコーダの音量は大きめにしておきましょう。しばらくすると，「LOADING PRO」とメッセージが出て，その後 「ＯＫ」が表示されます。これで，テープからの呼び出しが終わりました。リストを画面に出してみると，さっき打ち込んだプログラムが，そのとおりに出てきます。



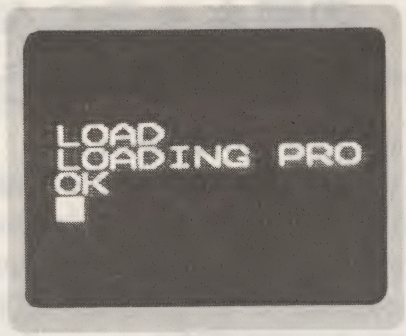


写真 3

さて，うまくいかなかった人はいますか？ そういう人は，もう１度線のつなぎ方をよく確かめて，音量や音質をいろいろ変えたり，始めからやり直してみたりしてみてください。それでもだめなら，ほかのテープレコーダに替えてみましょう。コンピュータとテープレコーダには，相性の良いものと悪いものがあり，相性の悪いものだと，いくら試してもうまくいかないことがあるからです。



第2章
テクニック編



1. 内蔵キャラクタを画面に出すもう1つの方法

●キャラクタにいろんな格好をさせる

さて，前にいろいろなキャラクタを画面上で動かすのに，DEF MOVE, POSITION, MOVEという3 つの命令を使いましたが，画面に内蔵キャラクタを出す方法がもう1つあります。使う命令は，DEF SPRITEとSPRITE という2つの命令です。まず次のプログラムを打ち込んでください。あっ，その前に，[N][E][W][RETURN] としておくのを忘れないように。前のプログラムが残っているとエラーの原因となることがあるのです。

```
100  CLS:CGSET 1, 1                            ← 画面をきれいにして,色を1の組にする
110  SPRITE ON                                 ← 内蔵キャラクタを表示する準備をする
120  DEF SPRITE Ø, (Ø, 1, Ø, Ø, Ø)             ← マリオのWALK1パターンに決める
=CHR$(Ø)+CHR$(1)+CHR$(2)+CHR
$(3)
130  SPRITE Ø, 12Ø, 12Ø                        ← マリオを画面にだす
```

リスト 3

プログラムを打ち込んだらRUNしてみてください。画面には，今にも歩き出しそうな左向きのマリオが出てきましたね。でも，今度のマリオは止まったままです。ここでちょっと，付録1のキャラクタコード表(P.86) を見てください。「マリオ(WALK 1)」というのがありますね。これが，今見ているマリオです。では今度は，「マリオ(WALK 2)」を画面に出してみましょう。

リスト３の120行に，CHR$（　）という部分が４つあり，（　）の中の数字は「∅」，「１」,「２」,「３」になっていますね。この４つの数字をそれぞれ「４」,「５」,「６」,「７」に直して，またRUNしてみてください。これが「マリオ（WALK-2)」です。さらに，「８」,「９」,「1∅」,「11」に直してRUNすると「マリオ（W-ALK3)」になります。このように，DEF SPRITEとSPRITE 命令を使うと，止まってはいますが，それぞれのキャラクタに，いろいろな格好(かっこう)をさせて，画面に出すことができるのです。

もう少し詳(くわ)しく説明すると，

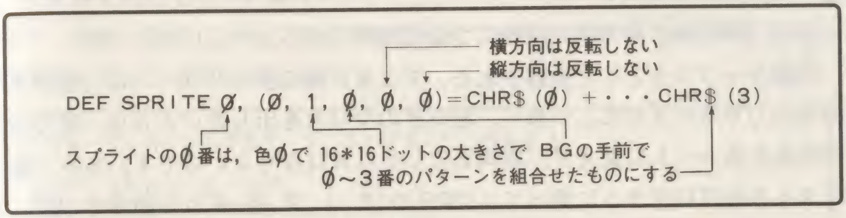


図　6

ということになります。

「色∅で」と「BGの手前で」は，DEF MOVE のときと同じです。また,「16ドット＊16ドットで」と「∅～３番のCHR$（　）を組み合わせたもの」というのは，図で説明すると，図７のようになります。

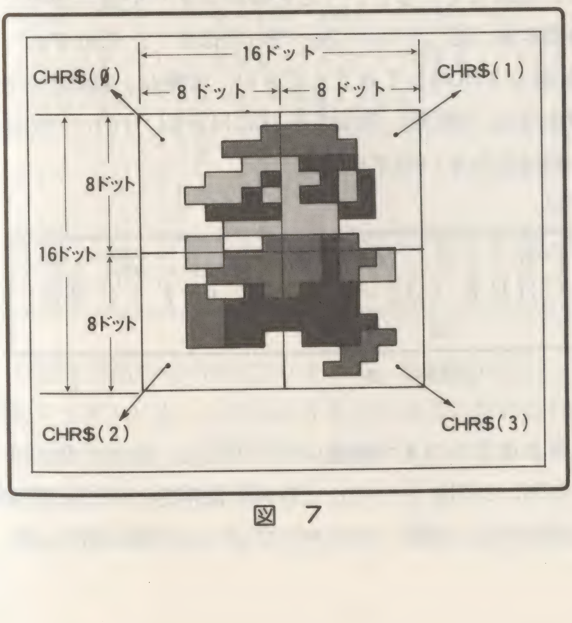


図　7

つまり，マリオ１人のパターンは，文字(CHR$）４つ分を図のように並べたものなのです。CHR$ １つは，絵を組み合わせるための部品のようなものです。そして，CHR$ ４つを組み合わせたパターンは，画面のドットにすると，16ドット＊16ドット分と同じ大きさになり，

ＣＨＲ＄ ４文字のとき──→16ドット＊16ドット（１）

ＣＨＲ＄ １文字のとき──→８ドット＊８ドット（Ø）

ということになるのです。

ここでひとつ実験してみましょう。120行の「16ドット＊16ドットで」(１)を,「８ドット＊８ドットで」(Ø) に直して，RUNしてみてください。マリオの顔だけしか，画面に出てきませんね。つまり，ＣＨＲ＄(Ø)だけしか出ていないということなのです。

●マリオを反転させる

次に，横方向と縦方向の反転について説明しましょう。

内蔵キャラクタとして登録された，マリオが横に歩くパターンは，左向きのものしかありません。つまり，右向きのマリオを出したいときは，横方向の反転をØから１に変えて，右向きにしなければならないのです。もう一度リスト３のプログラムに戻って，120行の(Ø,１,Ø,Ø,Ø) の部分を,(Ø,１,Ø,１,Ø)に直してRUNしてみてください。これは「横方向に反転する」ということですから，右向きのマリオが出てくるはずです。ところが，マリオは２つになってしまって，ちゃんと出ませんね。つまり,「横方向に反転する」は，ＣＨＲ＄４つを組み合わせたパターン全体を反転するのではなく，ＣＨＲ＄を１つずつ別々に反転しているということなのです。では，さらにCHR$(　)の(　)の中の数字を,「Ø」,「１」,「２」,「３」ではなく，それぞれ「１」,「Ø」,「３」,「２」に直してRUNしてみてください。今度は，右向きのマリオになりました。このように,「反転」をしたら,「ＣＨＲ＄(　)の(　)の中の数字」を入れ替えなければならないのです。

縦方向に反転するときは，

```
120 DEF SPRITE Ø,(Ø,1,Ø,Ø,1)
=CHR$(2)+CHR$(3)+CHR$(Ø)+CHR
$(1)
```

リスト　４

となります。

さて，これで内蔵キャラクタを１つずつ画面に出す方法と，向きを変える方法がわかりましたね。なお，CHR$(　)の(　)の中の数字をいくつにすると何のパターンが出てくるのかは，付録１のキャラクタコード表に出ていま

す(P.86参照)。

●組み合わせは自由

ここらで，ちょっとイタズラをしてみましょう。リスト3のプログラムどおりに戻してから，CHR$ の(2)を(30)に，(3)を(31)に直してRUN してみてください。何が出てきましたか？とんでもないマリオ(?)が出てきましたね。つまり，DEF SPRITE命令は，CHR$ を組み合わせる命令で，CHR$ の何を組み合わせるのかは自由なのです。

ではまとめとして，付録1のキャラクタコード表にあるすべてのキャラクタを，順番に画面に出すプログラムを載せておきますから，興味のある人は打ち込んでしてみてください。

```
100 CLS:CGSET 1,1:A$="CHR$(
)"
110 SPRITE ON
120 COLOR 15,3,3:COLOR 17,3,
3:COLOR 15,7,3:COLOR 17,7,3
130 LOCATE 8,3:PRINT A$:LOCA
TE 12,5:PRINT"カラ"
140 LOCATE 8,7:PRINT A$:LOCA
TE 10,9:PRINT"マデノ パターン"
150 LOCATE 9,17:PRINT"スペースキー
ヲオスト"
160 LOCATE 8,19:PRINT"ツギノパター
ンガ デマス"
170 FOR I=0 TO 255 STEP 4
180 DEF SPRITE 0,(0,1,0,0,0)
=CHR$(I)+CHR$(I+1)+CHR$(I+2)
+CHR$(I+3)
190 SPRITE 0,120,120
200 LOCATE 14,3:PRINT RIGHT$
("   "+STR$(I),3)
210 LOCATE 14,7:PRINT RIGHT$
("   "+STR$(I+3),3)
220 PAUSE 10
230 IF INKEY$<>" " THEN 230
240 NEXT
250 GOTO 170
```



リスト 5

CHR$ (208)～CHR$ (255) までは，CHR$1つで1パターンのものや，CHR$8つで1パターンのものなどがあることがわかりますね。なお，プログラムを止めるときは，STOP キーを押してください。

●プログラミングもいろいろ

さっき順番に出してみたマリオを，同時に3つ画面に出してみましょう。

```
100 CLS:CGSET 1,1
110 SPRITE ON
120 DEF SPRITE Ø,(Ø,1,Ø,Ø,Ø)
=CHR$(Ø)+CHR$(1)+CHR$(2)+CHR
$(3)
130 DEF SPRITE 1,(Ø,1,Ø,Ø,Ø)
=CHR$(4)+CHR$(5)+CHR$(6)+CHR
$(7)
140 DEF SPRITE 2,(Ø,1,Ø,Ø,Ø)
=CHR$(8)+CHR$(9)+CHR$(1Ø)+CH
R$(11)
150 SPRITE Ø,12Ø,1ØØ
160 SPRITE 1,12Ø,12Ø
170 SPRITE 2,12Ø,14Ø
```

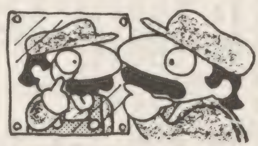


リスト 6

このプログラムでは，120と150，130と160，140と170行が，それぞれペアになっていて，DEF SPRITEのØ，1，2番を決めて，画面に出しています。

でもこのプログラム，ずいぶん同じような行が多いですね。次に載せるプログラムは，リスト6のプログラムとまったく同じ働きをするものですが，

```
100 CLS:CGSET 1,1
110 SPRITE ON
120 FOR I=Ø TO 2
130 DEF SPRITE I,(Ø,1,Ø,Ø,Ø)
=CHR$(I*4)+CHR$(I*4+1)+CHR$(
I*4+2)+CHR$(I*4+3)
140 SPRITE I,12Ø,1ØØ+2Ø*I
150 NEXT
```



リスト 7

どうですか？　だいぶ短くなったでしょう。このように，結果は同じでも，やり方が何通りもある場合があります。プログラミングのテクニックをマスターしたい人は，この2つを見比べて研究してみてください。

2. 内蔵キャラクタを動かす

●パターンを順番に出せば歩き出す

前項では動かなかったキャラクタに，いよいよ命を吹き込んでみましょう。

付録1のキャラクタコード表を見ると，CHR$ Ø～3を組み合わせたマリオは，WALK 1とありますね。同じように4～7，8～11はそれぞれWALK 2，WALK 3となっています。じつは，このWALK 1～3という3つのキャ

ラクタパターンは，マリオが歩いている様子を，連続写真で撮ったようなものです。

次のプログラムは，WALK 1→WALK 2→WALK 3→WALK 1→WALK 2→……というように，連続してパターンが見えるようにしたものです。

```
100 CLS:CGSET 1,1                          ← 画面をきれいにして，色を1の組にする。
110 SPRITE ON                              ← 内蔵キャラクタを表示する準備をする。
120 P=Ø:X=120:Y=120                        ← Pはパターン切替え・XとYは座標。
130 DEF SPRITE Ø,(Ø,1,Ø,Ø,Ø)               ← マリオのパターンを，順番にWALK1～3に決
=CHR$(P)+CHR$(P+1)+CHR$(P+2)                 める。
+CHR$(P+3)
140 SPRITE Ø,X,Y                           ← マリオを画面に出す。
150 P=P+4:IF P>8 THEN P=Ø                  ← Pを，0→4→8→0→4・・・と変化させる。
160 PAUSE 7                                ← 「7」の間，画面をそのままにする。
170 GOTO 13Ø                               ← 130行に行く。
```

リスト 8

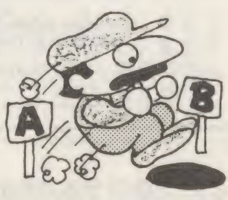
どうです？　ちゃんとマリオが歩いているように見えるでしょう。

では，150行の「PAUSE 7」を,「PAUSE 3Ø」に直してまたRUN してみてください。今度は，さっきよりマリオの動き方がゆっくりになりましたから，WALK 1～3の3パターンが順番に出てきているのが，よく見えますね。

●SPRITEで歩く場所を決める。

これで，マリオが歩いている様子だけはできましたが，まだマリオは画面のまんなかにいるままです。今度は，WALK 1～3 のパターンを順に出しながら，表示する場所を少しずつ左にずらしていくようにしてみましょう。

```
100 CLS:CGSET 1,1
110 SPRITE ON:P=Ø:Y=12Ø
120 FOR X=24Ø TO Ø STEP -4
130 DEF SPRITE Ø,(Ø,1,Ø,Ø,Ø)
=CHR$(P)+CHR$(P+1)+CHR$(P+2)
+CHR$(P+3)
140 SPRITE Ø,X,Y
150 P=P+4:IF P>8 THEN P=Ø
160 PAUSE 5
170 NEXT
180 GOTO 12Ø
```



リスト 9

画面のどこに出すかは，140行のSPRITE命令で決めています。DEF MOVE命令を使ったときは，動き始める位置をPOSITIONという命令を使って決めましたが，DEF SPRITE 命令の場合はSPRITE 命令を使って，画面のどの位置に出すかを決めます。このとき，DEF MOVEのときのように，「Ø番は」のØも一致させなければなりません。また，この数字はØ～7の8つが使えま

すから，同時に8つのキャラクタを画面に出せるわけです。

前項(ぜんこう)で，3種類のマリオのパターンを，同時に画面に出しましたが，あのときはDEF SPRITEのØ～2番を使っていました。

```
DEF SPRITE Ø, (Ø, 1, Ø, ・・・・・
           |
     ┌─────┴──────── 同じ数字にする（Ø～7）
     |
SPRITE Ø, X, Y
```

図 8

さて，リスト9のプログラムでは，マリオの横方向の位置を決めるのに，「X」を使って，順に変化させていますが，縦(たて)方向の位置は110行の「Y＝12Ø」で決めています。この「12Ø」をØ～255の範囲(はんい)でいろいろ変えると，マリオが画面の上の方を歩いたり，下の方を歩いたりしますから，試してみてください。

●右向きに歩かせよう。

これでマリオが歩きましたから，今度は右向きに歩かせてみます。これには前にやった，マリオを横方向に反転させる（右を向かせる）のを組み合わせればよいわけです。リスト10はリスト9のプログラムのうち，120行と130行を変えたものです。

＊100～110行と140～180行はリスト9と同じものを打ち込む

```
120 FOR X=Ø TO 240 STEP 4
130 DEF SPRITE Ø, (Ø, 1, Ø, 1, Ø)
=CHR$(P+1)+CHR$(P)+CHR$(P+3)
+CHR$(P+2)
```

リスト10

このプログラムをRUNすると，マリオは右に歩きます。つまり，横方向に反転させたパターンを，右方向にずらしながら画面に出しているわけです。

それではまとめとして，次のプログラムを載(の)せておきましょう。RUNすると，マリオが画面のまんなかから左に向かって歩き出し，端(はし)まで行くと方向を変えます。

```
100 CLS:CGSET 1,1
110 SPRITE ON
120 P=Ø:X=12Ø:Y=12Ø:AX=-4
13Ø '-----
14Ø IF AX<Ø THEN DEF SPRITE
Ø,(Ø,1,Ø,Ø,Ø)=CHR$(P)+CHR$(P
+1)+CHR$(P+2)+CHR$(P+3)
15Ø IF AX>Ø THEN DEF SPRITE
Ø,(Ø,1,Ø,1,Ø)=CHR$(P+1)+CHR$
(P)+CHR$(P+3)+CHR$(P+2)
16Ø SPRITE Ø,X,Y
17Ø IF X=Ø OR X=24Ø THEN AX=
-AX
18Ø X=X+AX
19Ø P=P+4:IF P>8 THEN P=Ø
2ØØ PAUSE 5
21Ø GOTO 14Ø
```



リスト 11

3. コントローラーでキャラクタを動かす

今までは，キャラクタが動かなかったり，動いても勝手に動いていましたが，今度は，コントローラで自由に動かせるようにしましょう。

●まずは動かしてみよう。

すぐにやってみたいでしょうから，さっそくプログラムを作ります。

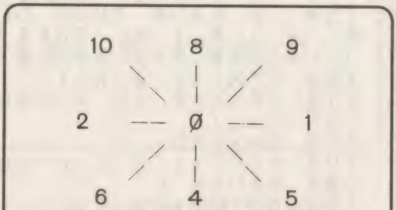
```
1ØØ CLS:CGSET 1,1                  ← 画面をきれいにする
11Ø SPRITE ON                      ← 内蔵キャラクタを表示する準備をする
12Ø P=96:X=12Ø:Y=12Ø               ← Pはパターン切替え・XとYは座標
13Ø '-----
2ØØ AX=Ø:K=STICK(Ø):IF K=Ø G ← コントローラーが何も押されなかったら，ずっとこ
OTO 2ØØ                            の行を実行し続ける
21Ø IF K<>1 OR X>=24Ø GOTO 2 ← コントローラーの右キーが押されたのでなかったら
4Ø                                 240行に行く
22Ø  DEF SPRITE Ø,(Ø,1,Ø,1,Ø ← スプライトの0番を，右向きのペンペンのパターン
)=CHR$(P+1)+CHR$(P)+CHR$(P+3       に決める
)+CHR$(P+2)
23Ø  AX=4                          ← 横方向にずらすのを，右に4ドットとする
24Ø IF K<>2 OR X<=Ø   GOTO 3 ← コントローラーの左キーが押されたのでなかったら
ØØ                                 300行に行く
25Ø  DEF SPRITE Ø,(Ø,1,Ø,Ø,Ø ← スプライトの0番を，左向きのペンペンのパターン
)=CHR$(P)+CHR$(P+1)+CHR$(P+2       に決める
)+CHR$(P+3)
26Ø  AX=-4                         ← 横方向にずらすのを，左に4ドットとする
3ØØ SPRITE Ø,X,Y                   ← スプライトの0番（ペンペン）を画面に出す
31Ø X=X+AX                         ← ペンペンをAXの分だけ横にずらす
32Ø P=196-P                        ← Pを96→100→96→100→と変化させる
33Ø PAUSE 4                        ← 4の間，画面をそのままにする
34Ø GOTO 2ØØ                       ← 200行に行く
```

リスト 12

このプログラムは，コントローラ1の左右のキーを押すと，ペンペンが左右に歩くものです。これが，コントローラでキャラクタを動かす一番基本的な形です。

●方向と数字の関係

プログラムの200行を見てください。STICK（Ø）というのがありますね。このSTICK（Ø）というのは，コントローラの方向キーを押すと，押された方向にしたがっていろいろな数字になるのです。方向と数字の関係は図9のようになっています。



コントローラの方向キーが押されたとき
STICKに入る数字

図　9

上下左右のほかにナナメもありますから，全部で8方向を判定することができ，方向キーが何も押されていないと，STICK（Ø）はØとなります。リスト12のプログラムでは，右と左だけですから，数字は1と2だけ判定すればいいことになります。プログラム中210行が右の判定，240行が左の判定ですから参考にしてください。

では，200行のSTICK（Ø）の（Ø）を（1）に直してもう1度RUN してみてください。今度は，コントローラ1の方向キーを押しても，ペンペンは動きませんね。では，コントローラ2の方向キーを押してみてください。今度はめでたくペンペンが動きました。つまり，

　コントローラ1…… STICK（Ø）
　コントローラ2…… STICK（1）

ということです。

●速度と数字の関係

それではまた，STICK（Ø）に戻してください。そして，330行にあるPAUSE 4をPAUSE 1に直して，RUNしてみましょう。ペンペンはどうなりましたか？　ずいぶん速く動くようになりましたね。「PAUSE　数字」は，この「数字」の分だけ画面をそのままにする，という働きがあります。この数字が大きいほど遅く，小さいほど速くなります。一番速いのは「Ø」です。あまり大きな数字にすると，ほとんど動かなくなってしまいますから注意しま

しょう。

もう１つ，違う実験をしてみましょう。330行はPAUSE 4に戻しておいてください。そして，今度は230行のAX＝４をAX＝1に，260行のAX＝－４をAX＝－１に直して，RUNしてみてください。330行はもとどおりPAUSE 4に直したのに，ペンペンはずいぶんゆっくり動くようになってしまいました。これは，横方向にずらしながら画面に出すときの，ずらす数値を４から１に変えたためなのです。つまり，¼の速さに変わったわけですね。その分だけ，ペンペンの動きが細かくなって，よりかわいらしくなったと思いませんか？

●数字を応用すると，風が吹く!?

次に，260行のAX＝－１はそのままで，230行のAX＝１だけをAX＝４に直して，RUNしてみてください。そして，右・左と動かしてみると，右に行くときは速く，左へ行くときは遅くなります。まるで左から風が吹いているようになります。でも，まだちょっと不自然ですから，思いきって大改造してしまいましょう。

```
100 CLS:CGSET 1,1
110 SPRITE ON
120 P=96:X=120:Y=120
130 '-----
200 AX=0:F=2:K=STICK(0):IF K
=0 THEN P=100:GOTO 230
210 IF K=1 AND X<240 THEN AX
=3:F=1
220 IF K=2 AND X>0   THEN AX
=-2
230 IF F=1 THEN DEF SPRITE 0
,(0,1,0,1,0)=CHR$(P+1)+CHR$(
P)+CHR$(P+3)+CHR$(P+2)
240 IF F=2 THEN DEF SPRITE 0
,(0,1,0,0,0)=CHR$(P)+CHR$(P+
1)+CHR$(P+2)+CHR$(P+3)
300 SPRITE 0,X,Y
310 X=X+AX:IF X<240 THEN X=X
+1
320 IF K<>0 THEN P=196-P
330 PAUSE 2
340 GOTO 200
```



リスト 13

どうです？ いかにも風が吹いているように見えるでしょう。プログラムというのは，その一部を変えたり，応用したりすると，いろいろ画面が変化するものですから，みなさんも自分でどんどん改造して試してみましょう。

エラーが出たぐらいで，ファミリーコンピュータは壊れません。エラーが出たら，そのときに考えればいいのですから，どんどんプログラムをいじってみることです。

●便利プログラムをひとつ…

では，まとめとして，プログラムをひとつ紹介しましょう。

```
100 CLS:CGSET 1,1
110 SPRITE ON
120 X=120:Y=120:F=1
130 P=0:P1=P:P2=8
140 '-----
200 AX=0:AY=0
210 K=STICK(0):IF K=0 GOTO 2
10
220  IF K=1 OR K=9 OR K=5  T
HEN AX=4 :F=1
230  IF K=2 OR K=6 OR K=10 T
HEN AX=-4:F=-1
240  IF K=4 OR K=5 OR K=6  T
HEN AY=4
250  IF K=8 OR K=9 OR K=10 T
HEN AY=-4
260 IF F=1  THEN DEF SPRITE
0,(0,1,0,1,0)=CHR$(P+1)+CHR$
(P)+CHR$(P+3)+CHR$(P+2)
270 IF F=-1 THEN DEF SPRITE
0,(0,1,0,0,0)=CHR$(P)+CHR$(P
+1)+CHR$(P+2)+CHR$(P+3)
280 X=X+AX
290  IF X>240 THEN X=0
300  IF X<0   THEN X=240
310 Y=Y+AY
320  IF Y>240 THEN Y=0
330  IF Y<0   THEN Y=240
340 SPRITE 0,X,Y
350 P=P+4:IF P>P2 THEN P=P1
360 PAUSE 3
370 GOTO 200
```

リスト 14

ちょっと長いかもしれませんが，これはとても便利なプログラムです。R-UNすると，マリオを8方向に歩かせることができます。画面の端っこまで行くと，今度は反対側から出てくるようになっています。そして，「便利な」と言ったのは，130行の数字を2ヶ所変えるだけで，すぐにペンペンやカニさんにすることもできるのです。どんな数字にするかは，表にしておきました

ので，利用してください。

130行を変更する時の，数字の組合わせ

```
130 P=0:P1=P:P2=8 ← マ リ オ
```

P	P2	
28 ・・・・・	36	← レ デ ィ
56 ・・・・・	60	← ファイターフライ
64 ・・・・・	68	← ア キ レ ス
88 ・・・・・	92	← ニ タ ニ タ
96 ・・・・・	100	← ペ ン ペ ン
112 ・・・・・	116	← ファイアーボール
144 ・・・・・	148	← ス ピ ナ ー
184 ・・・・・	188	← カ メ さ ん
192 ・・・・・	196	← カ ニ さ ん
200 ・・・・・	204	← ト リ さ ん

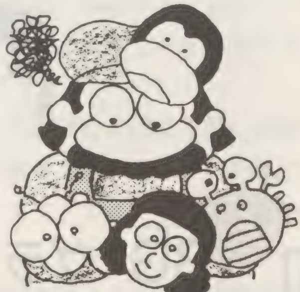


図 10

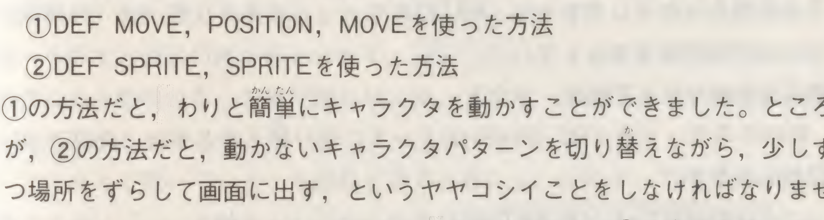
●プログラムを組みやすくするために。

ところで内蔵キャラクタを画面で動かすには，2つの方法がありましたね。

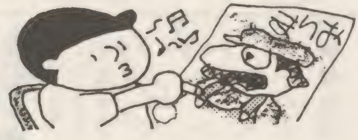
①DEF MOVE，POSITION，MOVEを使った方法

②DEF SPRITE，SPRITEを使った方法

①の方法だと，わりと簡単にキャラクタを動かすことができました。ところが，②の方法だと，動かないキャラクタパターンを切り替えながら，少しずつ場所をずらして画面に出す，というヤヤコシイことをしなければなりません。リスト12,13のプログラムを見ると，このヤヤコシイ②の方法を使っています。当然，「なんで簡単なほうを使わないんだ!?」と思っている人もたくさんいるでしょう。でも，動いているキャラクタが何かにぶつかった，何かを拾った，穴に落ちたなど，あとあとのことを考えると，②の方法の方が何かと便利なのです。だから，ちょっとヤヤコシイけれども，これからも②のDEF SPRITEの方を使って，話を進めていくことにします。



第3章 BG グラフィック編



1. BG グラフィックモードで絵を描く

さて，今まではバックに何も描いてない，まっ黒な画面でキャラクタを動かしていました。これからは，BG GRAPHICに描いてある絵を使っていろいろな判定をしたりしますから，BASICはちょっとお休みして，BG GRAPHICモードで絵を描きましょう。

●部品を組み立てて絵を「つくる」

BASICのモードからBG GRAPHICモードに切り替えるときは，次のように打ち込みます。

S Y S T E M RETURN

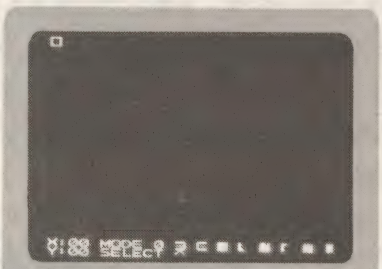


写真 4



画面が変わったら，[2]です。これでBG GRAPHIC(ビージーグラフィック)モードに入りました。

画面の右下に8個並んだ青いものが，絵を描く部品です。では，[CLR HOME]キーを押してみてください。1回押すごとにほかの部品が出てきますね。戻すときは，[SHIFT]キーを押しながら[CLR HOME]キーです。部品は全部で104個あります。

次に，[DEL]キーを押してみてください。すると，部品の下でピコピコしている「∧」マークが右に動きます。これでどの部品を使うかを選ぶのです。左に動かすときは[INS]キーを使います。

画面左上には，四角「□」がありますね。これは選んだ部品を置く位置を指示(しじ)するもので，矢印キーを押すと上下左右に動きます。矢印キーを押すときに，画面の左下を見ていると，数字が変わります。Xは，横方向に動かしたとき，Yは縦(たて)方向に動かしたときです。

この本の第6章にあるゲームで，BG GRAPHICを使ったものには表が付いていますが，それを描くときには，この数字をめやすにしてください。

では，[SPACE]キーを押してみてください。すると，「□」があった所に選んだ部品が出てきます。

今度は，色を変えてみましょう。[RETURN]キーを押してみてください。キーを押すと部品の色が変わります。[RETURN]キーを押すときに,「MODE」の右にある数字がØ～3まで変わります。つまり，全部で4つの色の組み合わせが選べるわけです。

まちがえて置いてしまった部品を消すときには，矢印キーで消したい部品のところに「□」を動かし,[D]キーを押してください。1つずつ消すのではなく全部まとめて消してしまいたいときは，

[ESC][↓][↓][↓][SPACE]

です。

●絵をSAVE, LOADする。

次は，描いた絵のカセットへの記録方法（SAVE）です。プログラムを記録したときのように，テープレコーダーがちゃんとつながっているかどうか確かめておいてください。そして，

[ESC][↓][↓][↓][↓][SPACE][S]

とキーインします。すると画面は写真5のようになります。なっていない人

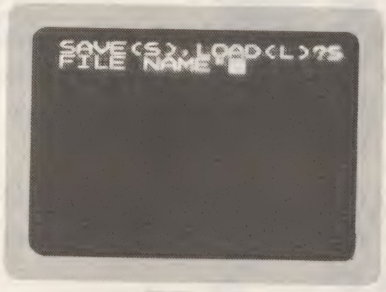


写真　5



は，STOPキーを押してから，やり直してください。ここで，プログラムのときに名前を付けたように，絵に名前(FILE名)を付けます。たとえば,「PICT」という名前にするときには，P I C Tと打ち込みます。そして，テープレコーダを録音状態にしてから，RETURNキーを押すと,「WRITING PICT」とメッセージが出て，記録が始まります。しばらくすると，画面がもとに戻ります。これで記録は終わりですから，テープレコーダを止めましょう。

テープにSAVEした絵を読み込む(LOAD)ときには，

ESC ↓ ↓ ↓ ↓ SPACE L

とキーインし，名前(FILE名)を打ち込みます。そしてテープレコーダを再生にしてから，RETURNキーを押してください。すると，「LOADING　名前」と出ます。これが出たら，読み込み始めたということです。「SKIP　名前」と出たときは，読み込んでいませんから，STOPキーを押してやり直してください。テープから読み込むときには,「名前」を省略してもかまいません。「LOADING　名前」が4回出て，全部読み終えると，読み込んだ絵が出てきますから，テープレコーダを止めておきましょう。

BG GRAPHICモードからBASICに戻るときは，

ESC STOP 1

とキーインしてください。

2. マリオの前には壁がある

これからは，BG GRAPHICで描いた絵を利用したいろいろな判定をしてみましょう。最初は，「マリオが壁にぶつかったらそれ以上進めなくなる」と



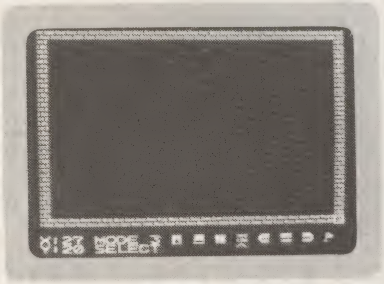


写真　6

いうものです。

●描いた絵をBASICで使う方法

まず，BG GRAPHICで画面の四方に壁を作ってください。部品は何を使ってもかまいませんが，ここでは付録2キャラクタテーブル(P.87)の「F 3」を「3」の色で使ったので，レンガの壁ができました。できあがったら BASIC に戻ってください。そして，[V][I][E][W][RETURN] と打ち込みます。すると，BG GRAPHICで描いた絵が，パッと出てきました。つまり，VIEWという命令を実行すると，BG GRAPHICの絵をBASICで使えるわけです。消すときは，[SHIFT] キーを押しながら [CLR HOME] キーです。

BG GRAPHICを使うようになったら，次のことに気をつけましょう。プログラムを一部直すときには，画面にリストを出して直すのですが，リストを出す前に [SHIFT] ＋ [CLR HOME] キーを押して，必ず画面をきれいにしなければなりません。そうしないと，リストに変な文字が混ざってしまって，エラーなどの原因になることがあります。

●文字が動く!? プログラム

さて，さっきBG GRAPHICで描いた壁はちゃんと残っていますね。それでは，次のプログラムを打ち込んでRUNしてみましょう。

このプログラムには，DEF SPRITEがありません。そのかわりに，PRINT P$ というのが入っています。そのP$には“A”という文字が入っているので（120行），画面では「A」という文字が４方向に動きまわるのです。これ

```
100 CLS:CGSET 1,1:VIEW
110 SPRITE ON
120 P$="A":X=13:Y=10
130 '-----
200 AX=0:AY=0
210 K=STICK(0):IF K=0 GOTO 2
10
220  IF K=1 THEN AX=1
230  IF K=2 THEN AX=-1
240  IF K=4 THEN AY=1
250  IF K=8 THEN AY=-1
260 K$=SCR$(X+AX,Y+AY)
270 LOCATE X,Y:PRINT " "
280 IF K$=" " THEN X=X+AX:Y=
Y+AY
290 LOCATE X,Y:PRINT P$
300 GOTO 200
```

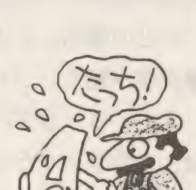


リスト 15

には，ちゃんと理由があるのですよ。

BG GRAPHIC(ビージーグラフィック)で絵を描くときには，部品を使いましたが，あの部品は１つで文字１つ分なのです。そして，260行にあるSCR$（　）※というのは，画面

```
100 CLS:CGSET 1,1:VIEW
110 SPRITE ON
120 P=0:X=13:Y=10:F=1
130 '-----
200 AX=0:AY=0
210 K=STICK(0):IF K=0 GOTO 2
10
220  IF K=1 THEN AX=1 :F=1
230  IF K=2 THEN AX=-1:F=2
240  IF K=4 THEN AY=1
250  IF K=8 THEN AY=-1
260 K$=SCR$(X+AX,Y+AY)
270 IF F=1 THEN DEF SPRITE 0
,(0,1,0,1,0)=CHR$(P+1)+CHR$(
P)+CHR$(P+3)+CHR$(P+2)
280 IF F=2 THEN DEF SPRITE 0
,(0,1,0,0,0)=CHR$(P)+CHR$(P+
1)+CHR$(P+2)+CHR$(P+3)
290 IF K$=" " THEN X=X+AX:Y=
Y+AY
300 SPRITE 0,X*8+12,Y*8+16
310 P=P+4:IF P>8 THEN P=0
320 PAUSE 5
330 GOTO 200
```



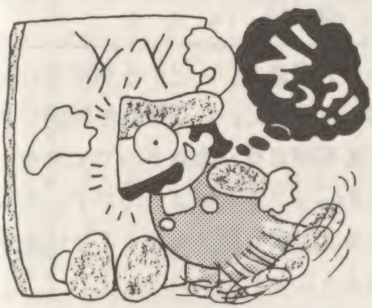
リスト 16

に何という文字が描いてあるかを見るものです。つまり，画面に描いてある絵を利用して，何かを判定するときには，文字単位で考えた方が楽なので，まず文字が動くプログラムにしてみたのです。

では，今度はちゃんとマリオを動かすことにしましょう(リスト16)。

全体のかたちは，「A」を動かしたものとほとんど変わりませんが，細かいところが違っているのがわかりますね。

さて，マリオを動かすと，頭の部分が少し壁にくい込みますね。これは，すでに説明したように，マリオのパターンは，文字を4つ組み合わせた大きさであることから起こることなのです。マリオの足の部分を基準に判定していますから，頭の部分は，少しはみ出してしまうのです。左右の壁にぶつかったときも，鼻が少し壁にくい込んでいます。



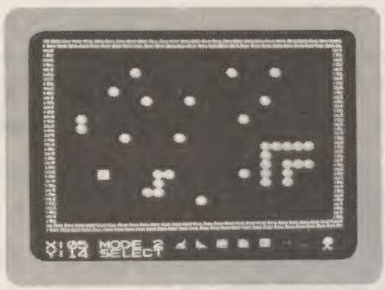


写真　7

●条件をつけて動かす

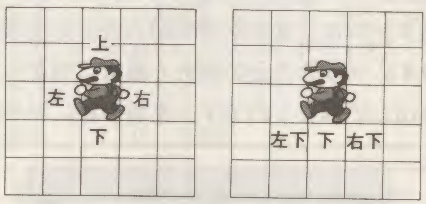
ではまた実験です。BG GRAPHICモードにして，適当に好きな場所に，好きな部品を置いてください。通路のようなものを作ってもいいし，ボールを点々と散らばしてもいいでしょう。ここでは，付録2のキャラクタテーブルの「G 7」を「2」の色で置いています。終わったら，またBASICに戻ってリスト16をRUNして，マリオをいろいろ動かしてみてください。
マリオは，あなたが置いた部品にぶつかると，それ以上進めないで，足だけを動かしていますね。つまり，このプログラムは，「コントローラの方向キーが押されたときに，その方向に何もなければ進み，何かあると進まない」というもので，キャラクタを使ったコントロール型ゲームを作るとき，基本的なパターンのひとつです。

※SCR$命令は，NS-HUBASIC V1.0にはありません。

3. マリオが穴に落ちる判定をする

BG GRAPHICを利用した判定の2つめです。今度は,「足の下に床があればそのまま進み,何もなくなったら下に落ちる」という判定をしてみましょう。

●右下、左下をチェック



※ 四角1つが1文字分の大きさです

図 11

さて,前項では,「進行方向に何かあったら進めない」という判定をしましたが,この場合,マリオを中心に図11の左のように,上下左右を判定していました。でも,今度は足の1歩先に床があるかどうかを見るのですから,図11の右のように,マリオの左下と右下をチェックすればよいのです。そして,落ちている最中には真下を見て,床があるまで落ち続けることになります。これをプログラムにすると,リスト17のようになります。

このプログラムは,BG GRAPHICで描いた絵によって判定するものですから,プログラムだけでRUNしてもうまく動きません。必ず,BG GRAPHICで何か絵を描いておいてください。そのとき,BG GRAPHICモードで,

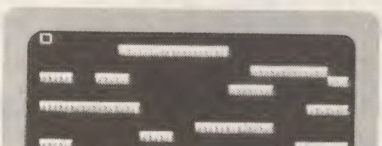


写真 8



```
100  CLS:CGSET 1,1:VIEW
110  SPRITE ON
120  P=0:X=13:Y=0:F=1
130  '-----
200  AX=0
210  K=STICK(0):IF K=0 GOTO 2
10
220   IF K=1 THEN AX=1 :F=1
230   IF K=2 THEN AX=-1:F=2
240  S=X+AX
250   IF S>27 THEN S=0
260   IF S<0  THEN S=27
270  K$=SCR$(S,Y+1):X=S
280  IF K$=" " GOTO 400
290  IF F=1 THEN DEF SPRITE 0
,(0,1,0,1,0)=CHR$(P+1)+CHR$(
P)+CHR$(P+3)+CHR$(P+2)
300  IF F=2 THEN DEF SPRITE 0
,(0,1,0,0,0)=CHR$(P)+CHR$(P+
1)+CHR$(P+2)+CHR$(P+3)
310  SPRITE 0,X*8+12,Y*8+15
320  P=P+4:IF P>8 THEN P=0
330  PAUSE 5:GOTO 200
340  '-----
400  IF Y>20 GOTO 200
410  FOR I=Y*8+16 TO Y*8+23
420   SPRITE 0,X*8+12,I
430  NEXT:Y=Y+1
440  IF SCR$(X,Y+1)=" " GOTO
400
450  GOTO 200
```



リスト 17

$\begin{cases} X=12 \\ Y=01 \end{cases}$ $\begin{cases} X=13 \\ Y=01 \end{cases}$ $\begin{cases} X=14 \\ Y=01 \end{cases}$

の３カ所には必ず何か部品を置くようにしてください。写真８では，付録(ふろく)２のキャラクタテーブルにある「Ｆ２」を使って，凍(こお)った床(ゆか)を作っています。

RUNしてみましたか？　マリオが左右に歩いて台のふちまで行くと，落ちて下の台に乗りますね。そして，画面の一番下まで落ちるともう動かなくなってしまいます。なお，このプログラムでは，画面の左右をつないでありますから，うまく台を作るとマリオは画面の端(はし)で消え，反対側から出て来るようになっています。

●ちょっと改造(かいぞう)すると……

さて，では少しこのプログラムを改造してみましょう。今のままだと，マ

リオが台のふちから落ちるときも，歩いている格好(かっこう)のままですから，落ちている間は，違(ちが)う格好をすることにしましょう。改造(かいぞう)といっても，次の1行を増やすだけです。

```
405 DEF SPRITE Ø, (Ø, 1, Ø, Ø, Ø)
=CHR$ (24) +CHR$ (25) +CHR$ (26) +
CHR$ (27)
```

リスト 18

●2つの判定(はんてい)をさせる

ではまとめとして，ひとつプログラムを紹介(しょうかい)しましょう。今度は，前に覚えた「何かにぶつかったらそれ以上進めなくなる」のと，今やった「床(ゆか)がなくなったら落ちる」というのを組み合わせたものです。この2つの判定をするプログラム，じつは，リスト17の270行を直し，265行を加えるだけでできてしまうのです。

```
265  K$=SCR$(S, Y): IF K$=" "THEN X=S
27Ø  K$=SCR$(S, Y+1)
```

リスト 19

さて，せっかく障害物(しょうがいぶつ)の判定ができるようになったのですから，今度は写真8のように，BG GRAPHICで作ってみた台の上に，障害物を置いてからRUNしてみてください。このへんまでくると，何だかもう本物のゲームのようになってきたと思いませんか？

4. レディの通り道には何がある

キャラクタが歩きまわって，何かを拾(ひろ)ったり取ったりするというのは，ゲームによくあるパターンですね。あの有名な「ロードランナー」も，金塊(きんかい)を拾ってまわるゲームです。ここでは，その「何かを拾う」というのをやってみましょう。

●部品を使い分ける

これも，BG GRAPHIC(ビージーグラフィック)で描いた絵を利用するということでは，前に作ったプログラムと似ています。プログラムは次のようになります。

```
100 CLS:CGSET 1,1:VIEW
110 SPRITE ON
120 P=28:X=13:Y=10:F=1
200 '-----
210 AX=0:AY=0
220 K=STICK(0):IF K=0 GOTO 2
00
230  IF K=1 THEN AX=1 :F=1
240  IF K=2 THEN AX=-1:F=2
250  IF K=4 THEN AY=1
260  IF K=8 THEN AY=-1
270 KK=ASC(SCR$(X+AX,Y+AY))
280 IF KK=32 THEN X=X+AX:Y=Y
+AY:GOTO 400
290 IF KK<>207 GOTO 400
300  LOCATE X+AX,Y+AY:PRINT
" "
310  X=X+AX:Y=Y+AY
400 '-----
410 IF F=1 THEN DEF SPRITE 0
,(0,1,0,1,0)=CHR$(P+1)+CHR$(
P)+CHR$(P+3)+CHR$(P+2)
420 IF F=2 THEN DEF SPRITE 0
,(0,1,0,0,0)=CHR$(P)+CHR$(P+
1)+CHR$(P+2)+CHR$(P+3)
430 SPRITE 0,X*8+12,Y*8+15
440 P=P+4:IF P>36 THEN P=28
450 PAUSE 5
460 GOTO 200
```



リスト 20



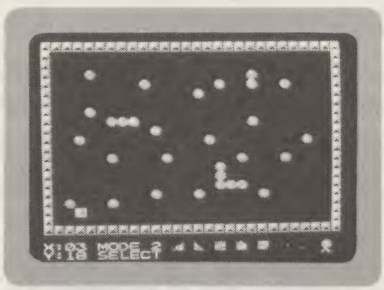


写真 9

このプログラムをRUNする前には，やはりBG GRAPHICモードで絵を描いておくのですが，それには，2つの約束があります。まず，四方には壁を作っておいてください。そして，壁に使う部品は，付録2キャラクタテーブル

の「G 7」以外のものを使ってください。「G 7」以外なら何でもかまいません。写真９では，「F Ø」を使っています。もう１つは，これから拾うものを適当に置くのですが，この拾うものは，「G 7」を使ってください。「G7」以外のものは，壁と同じに障害物と判定されてしまい，拾うことができません。では，絵が描けたらプログラムをRUNしてみましょう。

出てきたのはレディですね。このレディをいろいろ動かしてみてください。壁にぶつかるとレディはそれ以上進めません。ところが，「G 7」のボールの場所は通ることができ，通ったあとボールは消えます。つまり，レディがボールを拾った，ということになるのです。

●キャラクタテーブルを数字におきかえる方法

さて，リスト20の290行に，

```
290  1F  KK<>2Ø7  GOTO  4ØØ
```

というのがありますが，じつはこの「2Ø7」が「G 7」のボールのことなのです。プログラムが動いている人は，[STOP]キーで止めてください。そして，[SHIFT]＋[CLR HOME]（[CTRL]＋[L]でも同じです）で，画面をきれいにしてから，行番号はつけずに次のように打ってみてください。

[?][C][H][R][$][(][2][Ø][7][)][RETURN]

すると，画面には「G 7」のボールが出てきましたね。では，付録２のキャラクタテーブルを見てください。この表の中で，「E Ø」～「M 7」までのキャラクタには，184～255という番号が付いていて，BG GRAPHICモードにしなくても，画面に出すことができるのです。逆に，「E Ø」～「M 7」が画面に出て

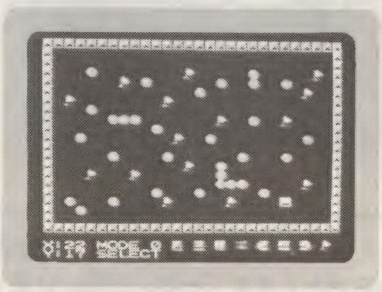


写真　10



いれば，その番号を利用して判定することができます。そして，このプログラムで使った「G７」(2Ø7)も，そのうちのひとつです。

●ボールを旗に変える

では，実験してみましょう。BG GRAPHICモードにして，「G７」のボールは残したままで，いくつか「F７」の旗を置いてください。今度は，この旗を拾えるように，プログラムを直してみましょう。まず，旗の番号が何番になるか，付録２キャラクタテーブルを見ながら数えてみてください。「EØ」が184ですから，右方向に185，186となり，「E７」が191です。一段下がって「FØ」が192で，と順に数えていくと「F７」の旗は199ですね。では，プログラムの290行を，

```
29Ø  1F  KK<>[199]  GOTO  4ØØ
```

と直して，RUNしてみましょう。今度はどうなりましたか？　レディは「F7」の旗を拾うようになりましたね。そして，さっきは拾えた「G７」のボールは，障害物になってしまいました。このように，290行を直すだけで，拾うものを変更することができるのです。

ではさらに，ボールと旗の両方を拾えるようにしてみましょう。これもやはり直すところは290行だけです。

```
29Ø  1F  KK<>199  AND  KK<>2Ø7  GOTO  4ØØ
```

このように，「KK<>番号」というものをANDでつないで並べると，同時に２種類の部品を判定できるようになります。

さて，以上が「物を拾う」というプログラムですが，壁に使う部品と拾う部品は，必ず違うものにしてください。そうしないと，壁が“拾う物”と判定されてしまい，壁を突き破って画面の外に出てしまいエラーになります。

5. 乱数を使う

●ワンパターンをくずせ

これまで，BG GRAPHICモードで絵を描き，それを利用していろいろな判

定をしてきました。そのうちの「物を拾う」ということについて，ちょっと考えてみましょう。

拾う物を置くときに，これまではBG GRAPHICモードにして自分で「ここと，あそこと，このへんにも1つ置いとこう」というようにしていました。

```
130 FOR I=1 TO 20
140 RX=2+RND(24):RY=2+RND(1
7)
150 IF SCR$(RX,RY)<>" " GOT
O 140
160 LOCATE RX,RY:PRINT CHR$
(207)
170 NEXT
```

リスト 21

この方法でプログラムをRUNすると,「いつも同じ場所に拾う物がある」ということになってしまいますね。そこで，「RUNするたびに拾う物のある場所が違う」というようにしてみましょう。こういうときには乱数というものを使うと便利です。

それでは，さっそくプログラムです。でも，これはリスト20のプログラムを利用したものなので，130～170行を追加するだけでいいのです(リスト21)。

さて，プログラムの打ち込みが終わったら，RUNする前に，BG GRAPHICモードにしてください。さっきいろいろと拾う物を置きましたが，四方の壁だけ残して，後は全部消しておきましょう。消すのは D キーでしたね，それが終わったら，BASICに戻してプログラムをRUNしてみてください。

RUNすると,「G 7」(CHR$(207))のボールが20個出てきましたね。では，STOP キーを押してプログラムの実行を止め，またRUNしてみてください。今度はさっきとは違う場所に20個のボールが出てきました。STOP してはRUNを繰り返すと，毎回ボールの場所が違っています。つまり乱数を使うと，プログラムは同じでも毎回違った画面をつくることができるわけです。

●ボールを使って試してみよう

それでは，またいろいろ実験してみましょう。

まず最初は，出てくるボールの数です。もうおわかりの人もいるでしょうが，プログラムの130行を直すと，ボールの数が変わります。たとえば10個にしたければ，

```
130  FOR  I＝1  TO  10
```

とすればいいのです。この数字を直すときは，あまり大きな数にするのはやめましょう。このあと別の実験をやる都合もありますから，50個以内にしてください。

今ボールは，だいたいまんべんなく壁の中いっぱいに散らばっています。今度は，左半分だけにボールが集まるようにしてみましょう。140 行を次のように直してRUNしてみてください。

```
140  RX＝2＋RND (12)：RY＝2＋RND (17)
```

何回RUNしても，ボールは画面の左半面にしか出てきません。次は，

```
140  RX＝ 14 ＋RND (12)：RY＝2＋RND (17)
```

にしてみましょう。これだと右半面だけになります。今度は，上半面だけにしてみましょう。

```
140  RX＝2＋RND (24)：RY＝2＋RND (8)
```

最後は，左半面と上半面を組み合わせてみましょう。

```
140  RX＝2＋RND (12)：RY＝2＋RND (8)
```

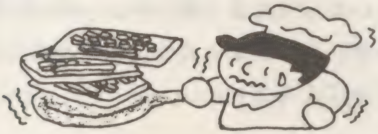
こうすると，ボールは左上 1/4 面にだけ出てきます。

いろいろやってみましたが，乱数を使うとプログラムに変化をつけることができます。ほかにも使いみちがたくさんありますが，その 1 例として，このプログラムの 160 行を次のように直してRUNしてみてください。

```
160  LOCATE RX, RY：PRINT CHR$ (199＋8＊RND (2))
```

こうすると，CHR$ (　) の中の数が199になったり207になったりし，画面には，旗が出たりボールが出たりするというわけなのです。

第4章
ハイ・テク編



1. 2人で内蔵キャラクタを動かす

今までは，コントローラ1だけを使い，1つのキャラクタを動かしていましたが，今度は，2人で遊べるように，コントローラ1，2を使って，2つのキャラクタを同時に動かしてみましょう。

●マリオとレディを2人で動かそう

2つのキャラクタを動かすといっても，1つのキャラクタだけを動かしていたときと，することは同じです。ただ，コントローラ1，2について，別々に2回やればいいのです。これをプログラムにすれば，リスト22のようになります。このプログラムをRUNするときも，BG GRAPHICで四方の壁だけ作っておいてください。部品は何を使ってもかまいません。

プログラムをRUNさせると，コントローラ1でマリオを，2でレディを別々に自由に動かすことができますね。さて，2人でコントローラを操作していると，まるでマリオとレディが同時に動いているように見えます。ところが，特殊なコンピュータでない限り，2つの物を同時に動かすなんてことはできないのです。リスト22のプログラムを見ても，コントローラ1でマリオを動かしたあと，2でレディを動かし，また頭に戻って1でマリオを動かし……というように，マリオとレディを交互に動かすようになっています。マリオとレディが同時に動いているように見えるのは，ファミリーコンピュ

ータが，命令をすばやく実行するためなのです。いろいろな物が４つも５つも画面を動きまわるゲーム，たとえば「マリオブラザース」や「ドンキーコング」なども，ひとつひとつを順番に少しずつ動かしているのです。

```
100  CLS:CGSET 1,1:VIEW
110  SPRITE ON
120  PA=Ø :XA=2 :YA=1Ø:FA=1
130  PB=28:XB=25:YB=1Ø:FB=2
2ØØ  '-----
21Ø  AX=Ø:AY=Ø
22Ø  K=STICK(Ø):IF K=Ø THEN P
AUSE 4:GOTO 4ØØ
23Ø   IF K=1 THEN AX=1 :FA=1
24Ø   IF K=2 THEN AX=-1:FA=2
25Ø   IF K=4 THEN AY=1
26Ø   IF K=8 THEN AY=-1
27Ø  K$=SCR$(XA+AX,YA+AY)
28Ø  IF K$=" " THEN XA=XA+AX:
YA=YA+AY
29Ø  IF FA=1 THEN DEF SPRITE
Ø,(Ø,1,Ø,1,Ø)=CHR$(PA+1)+CHR
$(PA)+CHR$(PA+3)+CHR$(PA+2)
3ØØ  IF FA=2 THEN DEF SPRITE
Ø,(Ø,1,Ø,Ø,Ø)=CHR$(PA)+CHR$(
PA+1)+CHR$(PA+2)+CHR$(PA+3)
31Ø  SPRITE Ø,XA*8+12:YA*8+15

32Ø  PA=PA+4:IF PA>8 THEN PA=
Ø
4ØØ  '-----
41Ø  AX=Ø:AY=Ø
42Ø  K=STICK(1):IF K=Ø THEN P
AUSE 4:GOTO 6ØØ
43Ø   IF K=1 THEN AX=1 :FB=1
44Ø   IF K=2 THEN AX=-1:FB=2
45Ø   IF K=4 THEN AY=1
46Ø   IF K=8 THEN AY=-1
47Ø  K$=SCR$(XB+AX,YB+AY)
48Ø  IF K$=" " THEN XB=XB+AX:
YB=YB+AY
49Ø  IF FB=1 THEN DEF SPRITE
1,(Ø,1,Ø,1,Ø)=CHR$(PB+1)+CHR
$(PB)+CHR$(PB+3)+CHR$(PB+2)
5ØØ  IF FB=2 THEN DEF SPRITE
1,(Ø,1,Ø,Ø,Ø)=CHR$(PB)+CHR$(
PB+1)+CHR$(PB+2)+CHR$(PB+3)
51Ø  SPRITE 1,XB*8+12,YB*8+15

52Ø  PB=PB+4:IF PB>36 THEN PB
=28
6ØØ  '-----
61Ø  GOTO 2ØØ
```

←コントローラー(1)の方(マリオ)の準備（120）

←コントローラー(2)の方(レディ)の準備（130）

コントローラー(1)でマリオを動かす（200〜320）

コントローラー(2)でレディを動かす（400〜520）

※BG GRAPHICモードで四方の壁を作ってからプログラムをRUNさせてください。

リスト 22

このあとも，２つ以上のものが画面の上を動くプログラムが出てきますが，考え方はすべて同じです。

●プログラムは早く，楽に打ち込みたい

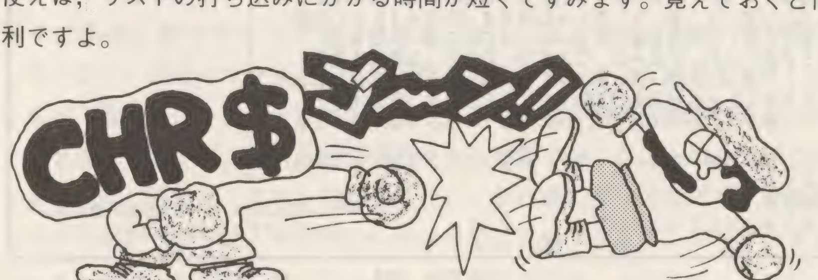
ところで，リスト22の200～320行と400～520行をよく見ると，数字が違うだけで後は同じ内容になっていますね。こういうプログラムを打ち込むとき，同じことを繰り返し打たないで，簡単にできる方法がありますので，そのコツをお教えしましょう。

まず，プログラムの最初から320行までを打ち込みます。ここまではふつうですが，ここからちょっと変わります。画面には200～320行が出ていますね。そこで矢印キーを使って，カーソルを200行の「2」に合わせ，[4] [RETURN] と打ちます。するとカーソルは，210行の「2」の上にきますから，また [4] [RETURN] と打てばいいのです。この操作を，290行まで繰り返したら，今度は300行の「3」の上にカーソルを合わせ，[5] [RETURN] と打ちます。同じようにして，320行まで繰り返したら，画面にリストを出してみてください。

どうですか？　200～320行と同じものが，400～520行にコピーされていますね。つまり，画面に出ているリストの行番号だけを直すと，その行をそっくりそのまま他の行にコピーできるのです。後は，400～520行の数字や文字を，本のリストどおりに直していけばいいのです。

この方法なら，打つときにずいぶん楽だし，まちがいも少なくなります。もっとも，コピーする行にまちがいがあれば，それも一緒にコピーされてしまいますが……。

なお，230～260行，290～390行のようなところでも使えますから，うまく使えば，リストの打ち込みにかかる時間が短くてすみます。覚えておくと便利ですよ。



●ぶつかったときに音を出す

では終わりに，ひとつおまけを付けましょう。リスト22のプログラムをちょっと直して，マリオとレディがぶつかったら，音が出るようにしてみましょう。605行として１行つけ加えてRUNしてみてください。今度は，マリオとレディがぶつかったときに，音がします。

```
6Ø5  IF  XA＝XB  AND  YA＝YB  THEN  BEEP
```

2. コンピュータが動かすキャラクタ

●相手を動かす

前項(ぜんこう)では，キャラクタ（マリオとレディ）を出し，両方とも，コントローラー1,2 で動かすのをやりましたが，今度は自分の動かすキャラクタに対して，相手はコントローラではなく，ファミリーコンピュータが動かすことにしましょう。相手の動き方にはいろいろなものがあるでしょうが，まず，簡単(かんたん)なものからやってみます。

```
100  CLS:CGSET 1,1:VIEW
110  SPRITE ON
120  PA=Ø  :XA=13:YA=19:FA=1      ←マリオの準備
130  PB=192:XB=13:YB=5  :BX=1     ←カニさんの準備
     ＊200〜320行は，リスト22と同じものを打ち込む
400  '-----
410  K$=SCR$(XB+BX,YB)
420  IF K$<>" " THEN BX=-BX
430  XB=XB+BX
440  DEF SPRITE 1,(3,1,Ø,Ø,Ø)
=CHR$(PB)+CHR$(PB+1)+CHR$(PB
+2)+CHR$(PB+3)
450  SPRITE 1,XB*8+12,YB*8+15

460  PB=PB+4:IF PB>196 THEN P
B=192
470  PAUSE 1
500  '-----
510  GOTO 200
```

（410〜470行：カニさんが横に動き，壁にぶつかったら，はねかえる）

※BG GRAPHICモードで四方の壁を作ってからプログラムをRUNさせてください。

リスト 23

このプログラムのうち，200～320行は，リスト22の200～320行と同じです。また，プログラムをＲＵＮする前に四方の壁(かべ)だけ BG GRAPHIC(ビージーグラフィック) でつくっておいてください。

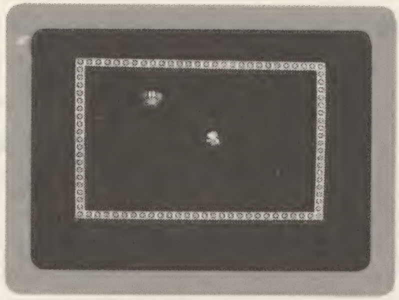


写真　11

さて，ＲＵＮするとカニさんが出てきて，左右に動き，壁にぶつかるとはね返って進みます。そして，コントローラ１では，マリオを上下左右に動かすことができ，マリオをどう動かしてもカニさんは同じ動きをし続けます。

ここでちょっとした実験もできます。BG GRAPHIC でカニさんの通り道（Y＝Ø5）に何か障害物(しょうがいぶつ)を置いてみてください。 部品は何でもかまいません。そしてＲＵＮするとカニさんはあなたの置いた障害物にぶつかるとはね返るようになります。

●相手の動きを右へ，左へ

では，次にカニさんの動くパターンを変えてみましょう。リスト24の 200 ～ 320 行はリスト23の 200 ～320 行と同じですが， 130 行は違(ちが)っていますから注意してください。またリスト23のときと同じように四方の壁だけ，BG GRAPHIC で作っておいてください。

今度のカニさんは，「壁にぶつかったら，右に曲がって進む」というパターンになっています。この場合の右とは，あなたから見て右ではなく，カニさんが進んでいる方向で見て右ということです。

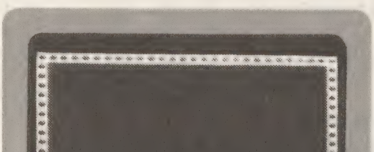
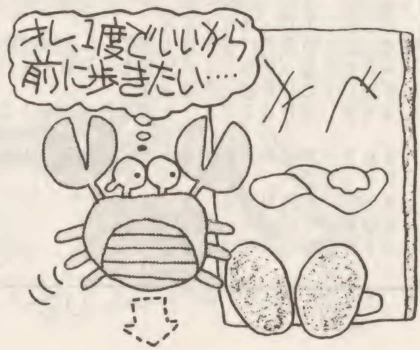


写真　12



```
100 CLS:CGSET 1,1:VIEW
110 SPRITE ON
120 PA=0  :XA=13:YA=19:FA=1        ←マリオの準備
130 PB=192:XB=13:YB=5              ←カニさんの準備
    *200～320行は，リスト22と同じものを打ち込む
400 '-----
410 BX=0:BY=0
420  IF DB=1 THEN BY=-1
430  IF DB=3 THEN BX=1
440  IF DB=5 THEN BY=1
450  IF DB=7 THEN BX=-1
460 K$=SCR$(XB+BX,YB+BY)
470 IF K$=" " THEN XB=XB+BX:
YB=YB+BY:GOTO 490
480 DB=DB+2:IF DB>7 THEN DB=
1
490 DEF SPRITE 1,(3,1,0,0,0)
=CHR$(PB)+CHR$(PB+1)+CHR$(PB
+2)+CHR$(PB+3)
500 SPRITE 1,XB*8+12,YB*8+15
510 PB=PB+4:IF PB>196 THEN P
B=192
520 PAUSE 1
600 '-----
610 GOTO 200
```

（420～610行）カニさんが進み、壁にぶつかったら、方向を右に変える

※BG GRAPHICモードで四方の壁を作ってからプログラムをRUNさせてください。

リスト 24

さて，480行に，

480 DB=DB+2：1F DB>7 THEN DB=1

とあうのがありますが，これを，

480 DB=DB [－] 2：IF DB [<1] THEN DB= [7]

というように直してＲＵＮしてみてください。今度は，カニさんは壁（かべ）にぶつかると，左に曲がるようになりました。

●相手の動く条件を増やす

今までやってみたカニさんの動きは，どれも「～になったら～する」という条件が１つだけでした。次のプログラムは，その条件を２つにしたものです。このプラグラムも，ＲＵＮする前に四方の壁をBG GRAPHIC（ビージーグラフィック）で作っておいてください。また，200～320行は，リスト23，24の200～320行と同じです(リスト25)。

ではＲＵＮしてみてください。乱数（らんすう）を使っているせいもあって，今度のカニさんはなかなか複雑（ふくざつ）な動きをしますね。それでも「～になったら～する」という条件は２つだけなのですよ。

さて，カニさんの動きをじっと見てみてください。今，カニさんは，ある

```
1ØØ CLS:CGSET 1,1:VIEW
11Ø SPRITE ON
12Ø PA=Ø   :XA=13:YA=19:FA=1        ←マリオの準備
13Ø PB=192:XB=13:YB=5  :DB=1        ←カニさんの準備
    *200～320行は，リスト22と同じものを打ち込む
4ØØ '-----
41Ø BX=Ø:BY=Ø
42Ø  IF DB=1 THEN BY=-1
43Ø  IF DB=3 THEN BX=1
44Ø  IF DB=5 THEN BY=1
45Ø  IF DB=7 THEN BX=-1
46Ø K$=SCR$(XB+BX,YB+BY)
47Ø IF K$=" " GOTO 53Ø
48Ø  IF DB=1 THEN DB=5 :GOTO
 55Ø
49Ø  IF DB=3 THEN DB=7 :GOTO
 55Ø
5ØØ  IF DB=5 THEN DB=1 :GOTO
 55Ø
51Ø  IF DB=7 THEN DB=3
52Ø  GOTO 55Ø
53Ø R=RND(2Ø):IF R>Ø THEN XB
=XB+BX:YB=YB+BY:GOTO 55Ø
54Ø DB=DB+2:IF DB>7 THEN DB=
1
55Ø DEF SPRITE 1,(3,1,Ø,Ø,Ø)
=CHR$(PB)+CHR$(PB+1)+CHR$(PB
+2)+CHR$(PB+3)
56Ø SPRITE 1,XB*8+12,YB*8+15
57Ø PB=PB+4:IF PB>196 THEN P
B=192
58Ø PAUSE 1
6ØØ '-----
61Ø GOTO 2ØØ
```

（480～520行）壁ではね返る
（530～540行）まっすぐ進むか，右に曲る
（410～580行）カニさんが動く

※BG GRAPHICモードで四方の壁を作ってからプログラムをRUNさせてください。

リスト 25

程度まっすぐ進むことが多いですね。では，プログラムの530行を次のように直してRUNし直して，またカニさんの動き方を見てみてください。

```
53Ø R=RND( 3 ):IF R>Ø THEN XB=
XB+BX:YB=YB+BY:GOTO 55Ø
```

今度は，カニさんはちょこちょこと進む方向を変えています。では，また530行を次のように直してRUNしてみてください。

```
53Ø R=RND( 5Ø ):IF R>Ø THEN XB
=XB+BX:YB=YB+BY:GOTO 55Ø
```

カニさんは，進む方向をあまり変えなくなってしまいました。

これは，まっすぐに進んでいる最中「時々向きを変える」というのを，「しょっちゅう向きを変える」や，「たまに向きを変える」にしてみたのです。このプログラムの場合，ＲＮＤ（　）の（　）の中の数字が小さいとしょっちゅう向きを変え，大きいとあまり向きを変えなくなるわけです。

カニさんの動きが，このくらい複雑(ふくざつ)になってくると，もうゲームの完成品まであとひと息(いき)という感じです。ゲームとして他に必要なのは，

①マリオがカニさんをつかまえたら得点を入れる。

②マリオがカニさんをつかまえたら別の場所にカニさんを出す。

③一定(いってい)時間たったら，ゲームを終わらせる。

くらいです。ついでに、あらかじめ障害物(しょうがいぶつ)を置いておいたり，カニさんをつかまえたらその位置に障害物を置くようにする，などの味付(あじつ)けをすれば，もう立派(りっぱ)なゲームといえるでしょう。

3. 相手と自分を組み合わせる

●ハエがマリオを追いかける

さっきはカニさんが出てきて，あなたの動きに関係なく動きまわっていたわけですが，今度は「ハエさんがマリオを追いかけて来る」というのをやってみましょう。

```
100 CLS:CGSET 1,1:VIEW
110 SPRITE ON
120 PA=0 :XA=13:YA=19:FA=1          ←マリオの準備
130 PB=56:XB=13:YB=5                ←ハエさんの準備
    *200～320行は，リスト22と同じものを打ち込む
400 '-----
410 R=R+1:IF R>1 THEN R=0
420 IF R<>0 GOTO 450
430 BX=SGN(XA-XB):BY=SGN(YA-
YB)
440 XB=XB+BX:YB=YB+BY
450 DEF SPRITE 1,(0,1,0,0,0)
=CHR$(PB)+CHR$(PB+1)+CHR$(PB
+2)+CHR$(PB+3)                      ハエさんが，マリオを追いかける
460 SPRITE 1,XB*8+12,YB*8+15

470 PB=PB+4:IF PB>60 THEN PB
=56
480 PAUSE 1                         ※BG GRAPHICモードで四方
500 '-----                            の壁を作ってからプログラム
510 GOTO 200                          をRUNさせてください。
```

リスト 26

このプログラムをRUNするときも，BG GRAPHICで四方の壁を作っておいてください。なお，プログラム中の200～320行は，リスト22～25の 200～320行と同じです。

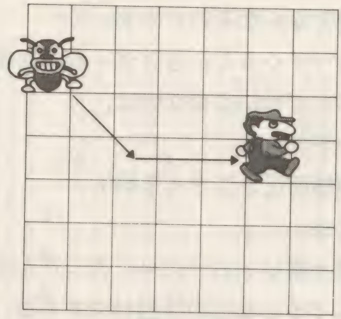


図 12

ハエさんがマリオを追いかけてくるコース

プログラムをRUNしたら，すぐにコントローラ1の方向キーで，マリオを動かしてみましょう。ハエさんはどんどん追いかけてきます。ハエさんにつかまらないようにするには，いつまでも逃げ続けなければなりません。これが「追いかけプログラム」のいちばん単純なものです。

●相手の速度を変える

今飛んでるハエさんはずいぶんすばやく追いかけて来ますね。プログラムをちょっと直して，ハエさんがもっとゆっくり追いかけて来るようにしてしまいましょう。プログラムの410行は，

```
410 R=R+1:IF R>1 THEN R=0
```

ですが，これを，

```
410 R=R+1:IF R>4 THEN R=0
```

に直してRUNしてみてください。ハエさんは，さっきよりずっとゆっくり追いかけて来ます。マリオがちょっとの間立ち止まっている余裕もあります。今直したところの数字を大きくするほど，ハエさんの追いかけて来るスピードを遅くすることができます。

逆に小さくすれば，ハエさんのスピードは速くなり，一番速いのは「0」にしたときです。

```
410 R=R+1:IF R>0 THEN R=0
```

RUNしてみましたか？　1度ハエさんに追いつかれたら，もう2度と逃げることはできませんね。画面で見ると，まるでマリオに羽が生えてしまったようです。ハエさんがあんまり速く追いかけすぎるのでこうなってしまったのですが，これではしょうがないですから，ここの数字は実質的には「1」が1番速いということになるのです。

今度は，今の410行を次のように直してRUNしてみてください。

```
410  R=RND(4)
```

ハエさんの動きをよく見ていると，さっきまではなめらかな動きで追いかけて来ていましたが，今度は止まったり追いかけたりギクシャクと追いかけてきます。これは，ハエさんが追いかけてくる速さに，乱数を使ったためにこんな動き方になったのです。乱数にはこんな使い方もあるのですよ。

さてこの場合も410行の中の数字を大きくすると，ハエさんはゆっくり追いかけてくるようになり，小さくすると速くなります。

```
410  R=RND ([2])
```

たとえば（2）くらいにしてみると，逃げるのはかなりきびしくなります。また，（1）にするとさっきもなってしまったように,「マリオに羽根が生えた」状態になってしまいます。

●条件をつけて追いかけさせる

では，追いかけプログラムの最後です。410行を次のように直してRUNしてください。

```
410  R=(XA+YA) MOD 2
```

今度はマリオを動かし続けるのではなく，時々立ち止まらせてみてください。すると，ある場所に立ち止まると，ハエさんはそのまま追いかけて来ますが，ある場所では，ハエさんが追いかけて来なくなりますね。これは，マリオの位置によって，ハエさんが追いかけて来たり来なかったり，というように直したからです。このように,「ハエさんがマリオを追いかける」というプログラムでも，ちょっと変えることによって，いろいろなパターンにすることができるのです。

4. 宇宙を創造してみよう

すでに「相手を出す」というのをやりましたね。これは相手も動き，自分も動くという，「2つのものを動かす」というプログラムだったわけです。そして，動かすもの2つのうち1つは，コントローラ1で動かすマリオでした。残りの1つが，ファミリーコンピュータの動かすカニさんだったりハエさんだったわけです。

●ミサイル発射！

さて，これからやってみようとしているのは，やっぱり「2つのものを動かす」プログラムの一種で，「宇宙船からミサイルを発射する」というものです。何だかピンとこないと思っている人もいるでしょうから，まずプログラムを紹介しましょう。このプログラムをRUNするときには，BG GRAPHICは必要ありません。

リスト 27

```
100 CLS:CGSET 1,1
110 SPRITE ON
120 PA=164:PB=209:XA=13:YA=1
0:FM=0
130 A1$=CHR$(PA)+CHR$(PA+1)+
CHR$(PA+2)+CHR$(PA+3)            ┐左向きの宇宙船の準備
140 A2$=CHR$(PA+1)+CHR$(PA)+
CHR$(PA+3)+CHR$(PA+2)            ┐右向きの宇宙船の準備
150 A3$=CHR$(PA+8)+CHR$(PA+9
)+CHR$(PA+10)+CHR$(PA+11)        ┐上向きの宇宙船の準備
160 A4$=CHR$(PA+10)+CHR$(PA+
11)+CHR$(PA+8)+CHR$(PA+9)        ┐下向きの宇宙船の準備
170 B1$=CHR$(PB):                ─横向き(左右兼用)のミサイルの準備
180 B2$=CHR$(PB+4)               ─縦向き(上下兼用)のミサイルの準備
200 '-----
210 AX=0:AY=0:DX=0:DY=0          ┐
220 K=STICK(0):IF K=0 THEN P     │
AUSE 2:GOTO 400                  │
230  KK=K                        │
240  IF K=1 THEN AX=1 :PA$=A     │
2$:DX=1                          │
250  IF K=2 THEN AX=-1:PA$=A     │
1$                               │
260  IF K=4 THEN AY=1 :PA$=A     │
4$:DY=1                          ├コントローラ(I)で，宇宙船を動かす
270  IF K=8 THEN AY=-1:PA$=A     │
3$                               │
280 XA=XA+AX:YA=YA+AY            │
290  IF XA>27 THEN XA=0          │
300  IF XA<0  THEN XA=27         │
310  IF YA>23 THEN YA=0          │
320  IF YA<0  THEN YA=23         │
```

```
330 DEF SPRITE Ø,(Ø,1,Ø,DX,D
Y)=PA$
340 SPRITE Ø,XA*8+12,YA*8+15

400 '-----
410 IF FM<>1 THEN PAUSE 1:GO
TO 5ØØ
420  NB=NB-1
430  IF NB<Ø OR XB>27 OR XB<
Ø OR YB>23 OR YB<Ø THEN FM=Ø     ―ミサイルが画面の端まで行ったら，消す
:SPRITE 1:GOTO 5ØØ
440  XB=XB+BX:YB=YB+BY
450  SPRITE 1,XB*8+16,YB*8+1     ―ミサイルを，そのまま進ませる
9
460  GOTO 6ØØ
5ØØ KG=STRIG(Ø):IF KG<>8 GOT
O 6ØØ
51Ø  NB=14:FM=1:XB=XA:YB=YA
52Ø  IF KK=1 THEN BX=2 :BY=Ø
 :PB$=B1$
53Ø  IF KK=2 THEN BX=-2:BY=Ø     ―コントローラー(I)のAボタンが押され
 :PB$=B1$                           たら，ミサイルを発射する
54Ø  IF KK=4 THEN BX=Ø :BY=2
 :PB$=B2$
55Ø  IF KK=8 THEN BX=Ø :BY=-
2:PB$=B2$
56Ø  DEF SPRITE 1,(3,Ø,Ø,Ø,Ø
)=PB$
6ØØ '-----
61Ø GOTO 2ØØ
```

宇宙船のキャラクタを4方向使っていることなどで，プログラムがちょっと長くなってしまいましたが，何ごとも練習ですから，あせらずにのんびり打ち込んでください。





写真 13

●プログラムの全体像をつかもう

このプログラムは，コントローラ1の方向キーで宇宙船を4方向に動かし，Aボタンを押すとミサイルを発射するようになっています。プログラムには何だかいろいろ説明がついていますが，大ざっぱにいえば，200～340行で宇宙船を動かし，400～560行でミサイルを動かしているということです。つまり，「2つのものを動かす」プログラムの一種だというわけです。

プログラムの1行1行で，何をやっているのかよくわからないという人も，

ちょっとリスト26と27を見比べてみてください。全体のカタチが何となく似ているでしょう。だいたいプログラムなんてこんなものなのです。まず「準備」があって，次に「何かをする部分」があって，また「何かをする部分」があって……というように，1つ1つの部分をつなぎ合わせて全体ができています。BASICを覚えようとしている人は，命令やテクニックを1つ1つ覚えるのではなく，まず全体のイメージをつかむことが大事です。……何だか話が理屈っぽくなってしまいました。リスト27に話を戻しましょう。

リスト27では，宇宙船にスターシップ，ミサイルにレーザー（2）のキャラクタを使っています。ところが，120行をちょっと直すと，宇宙船をスターキラー，ミサイルをレーザー（1）のキャラクタに変えられます。変えるときは，それぞれ次のような数字にしてください。

```
120 [PA=164] : [PB=209] : XA=13: YA=1
0 : FM=0

PA=164←スターシップ     PB=209←レーザー(2)
PA=152←スターキラー     PB=208←レーザー(1)
```

図 13

●ミサイルをボールに変えても原理は同じ

ここで，マリオがボールを投げるプログラムを紹介しておきましょう。このプログラムをRUNするときは，BG GRAPHIC で四方の壁だけ作っておいてください。

リスト 28

```
100 CLS:CGSET 1,1:VIEW
110 SPRITE ON
120 PA=0:XA=13:YA=10:FM=0
200 '-----
210 AX=0:AY=0
220 K=STICK(0):IF K=0 THEN PAUSE 4:GOTO 400
230  KK=K
240  IF K=1 THEN AX=1 :F=1
250  IF K=2 THEN AX=-1:F=2
260  IF K=4 THEN AY=1
270  IF K=8 THEN AY=-1
280 K$=SCR$(XA+AX,YA+AY)
290 IF K$=" " THEN XA=XA+AX:YA=YA+AY
300 IF F=1 THEN DEF SPRITE 0,(0,1,0,1,0)=CHR$(PA+1)+CHR$(PA)+CHR$(PA+3)+CHR$(PA+2)
310 IF F=2 THEN DEF SPRITE 0,(0,1,0,0,0)=CHR$(PA)+CHR$(PA+1)+CHR$(PA+2)+CHR$(PA+3)
320 SPRITE 0,XA*8+12,YA*8+15
```



```
330 PA=PA+4:IF PA>8 THEN PA=
Ø
4ØØ '-----
41Ø IF FM<>1 THEN PAUSE 1:GO
TO 5ØØ
42Ø  NB=NB-1
43Ø  IF NB<Ø OR XB>25 OR XB<
2 OR YB>19 OR YB<2 THEN FM=Ø
:SPRITE 1:GOTO 5ØØ
44Ø  XB=XB+BX:YB=YB+BY
45Ø  SPRITE 1,XB*8+16,YB*8+1
9
46Ø  GOTO 6ØØ
5ØØ KG=STRIG(Ø):IF KG<>8 GOT
O 6ØØ
51Ø  NB=1Ø:FM=1:XB=XA:YB=YA
52Ø  IF KK<4 GOTO 55Ø
53Ø   IF F=1 THEN XB=XB+1
54Ø   IF F=2 THEN XB=XB-1
55Ø  IF KK=1 THEN BX=1 :BY=Ø

56Ø  IF KK=2 THEN BX=-1:BY=Ø

57Ø  IF KK=4 THEN BX=Ø :BY=1

58Ø  IF KK=8 THEN BX=Ø :BY=-
1
59Ø  DEF SPRITE 1,(2,Ø,Ø,Ø,Ø
)=CHR$(223)
6ØØ '-----
61Ø GOTO 2ØØ
```

※BG GRAPHICモードで四方の壁を作ってからプログラムをRUNさせてください。



宇宙船のときは，宇宙船がちゃんと４方向に向くのでミサイルを発射できる方向が画面をひと目見てわかりました。今度のマリオでは，マリオが左右しか向きません。ですからボールを投げる方向は，マリオが最後に動いた方向になっています。また，上下に投げるときにも，ちゃんと手の位置からボールが飛んで行くようにくふうしてあります（520〜540行）。

さて，ボールを投げてみるとわかりますが，壁にぶつからなくてもボールが途中で消えてしまいますね。これは，ボールが文字10個分飛んだらそこで終わりにする，というプログラムになっているからです。もっと遠くまで飛ばすときは，510行の，

510　NB＝10：FM＝1：XB＝XA：YB＝YA

を，次のように変えてください。

510　NB＝[20]：FM＝1：XB＝XA：YB＝YA

この数字を大きくすれば，遠くまで飛ぶのですが,「24」にすれば必ず壁にぶつかるようになりますから，それ以上にしても意味がありません。また，ボ

ールが飛んでいる間は次のボールが投げられませんから，ゲームに利用するときなど，適当な数字にしておいた方がいいでしょう。

5. 自分も動いて 相手も動いて

●マリオ VS カニさん

第3章では，BG GRAPHICを利用して，壁にぶつかった，何かを拾った，などの判定をしました。ここでの相手は，壁や落ちているものなど，みんな動かないものでした。また，BG GRAPHICで置いたものだったので，画面のどの部分にどんな部品があるのかを知るための，SCR$ ( )という命令を使いました。でも今度は，判定の相手もあなたと同じように動いているものなのです。

次のプログラムは，リスト25をもとにして作り変えたものです。これで判定をしてみましょう。プログラムをRUNするときは，BG GRAPHICで四方の壁だけ作っておいてください。

リスト 29

```
100 CLS:CGSET 1,1:VIEW
110 SPRITE ON
120 PA=Ø  :XA=13:YA=19:F=1
130 PB=192:XB=13:YB=3 :DB=1
200 '-----
210 AX=Ø:AY=Ø
220 K=STICK(Ø):IF K=Ø THEN P
AUSE 4:GOTO 400
230  IF K=1 THEN AX=1 :F=1
240  IF K=2 THEN AX=-1:F=2
250  IF K=4 THEN AY=1
260  IF K=8 THEN AY=-1
270 K$=SCR$(XA+AX,YA+AY)
280 IF K$=" " THEN XA=XA+AX:
YA=YA+AY
290 IF F=1 THEN DEF SPRITE Ø
,(Ø,1,Ø,1,Ø)=CHR$(PA+1)+CHR$
(PA)+CHR$(PA+3)+CHR$(PA+2)
300 IF F=2 THEN DEF SPRITE Ø
,(Ø,1,Ø,Ø,Ø)=CHR$(PA)+CHR$(P
A+1)+CHR$(PA+2)+CHR$(PA+3)
310 SPRITE Ø,XA*8+12,YA*8+19

320 PA+PA+4:IF PA>8 THEN PA=
Ø
400 '-----
410 BX=Ø:BY=Ø
420  IF DB=1 THEN BY=-1
430  IF DB=3 THEN BX=1
440  IF DB=5 THEN BY=1
450  IF DB=7 THEN BX=-1
460 K$=SCR$(XB+BX,YB+BY)
```



```
470 IF K$=" " GOTO 530
480   IF DB=1 THEN DB=5 :GOTO
 550
490   IF DB=3 THEN DB=7 :GOTO
 550
500   IF DB=5 THEN DB=1 :GOTO
 550
510   IF DB=7 THEN DB=3
520  GOTO 550
530 R=RND(5):IF R>0 THEN XB=
XB+BX:YB=YB+BY:GOTO 550
540 DB=DB+2:IF DB>7 THEN DB=
1
550 DEF SPRITE 2,(3,1,0,0,0)
=CHR$(PB)+CHR$(PB+1)+CHR$(PB
+2)+CHR$(PB+3)
560 SPRITE 2,XB*8+12,YB*8+19

570 PB=388-PB
580 PAUSE 1
600 '-----
610 IF XA<>XB OR YA<>YB GOTO
 700
620  BEEP
700 '-----
710 GOTO 200
```

※BG GRAPHICモードで四方の壁を作ってからプログラムをRUNさせてください。



さて，プログラムをRUNしたらすぐにマリオを動かして，カニさんを追いかけさせてください。すると，うまくつかまえたときには，ピーと音がしますね。では，プログラムの610行と620行を見てみてください。リスト25のプログラムには，なかった部分です。ここでは，マリオとカニさんが画面の同じ場所にいたら音を出す，ということをしているのです。ここが，壁(かべ)や穴の判定(はんてい)とは違(ちが)うところです。

●いろいろ追加してみよう

実際のゲームでは，音を出すだけではなく，得点が表示(ひょうじ)されたり，相手が爆発(ばくはつ)したりするわけですね。

```
630  SC=SC+1:LOCATE 13,21:PRINT SC
```

リスト29に，この630行を追加すると，今度はマリオがカニさんをつかまえたとき，音がするだけでなく，画面下の方に数字が出るようになります。これは，つかまえた回数を数えているので，つかまえるたびに，その数字が1つずつ増えていくわけです。この630行を，

```
630  SC=SC+[10]:LOCATE 13,21:PRINT SC
```

のように直せば，数字の増えかたが10ずつになりますから,「カニさんを1回

つかまえると10点」ということになります。これがスコアのつけ方です。

では，もう少しプログラムを追加してみましょう。少し長いですが，次のようなプログラムを，リスト29に書き加えて，RUN してみてください。

```
640   SPRITE Ø
650   FOR I=1Ø TO 15:PC=176+4
*(I MOD 2)
660    DEF SPRITE 1,(3,1,Ø,Ø,
Ø)=CHR$(PC)+CHR$(PC+1)+CHR$(
PC+2)+CHR$(PC+3)
670    SPRITE 1,XA*8+12,YA*8+
19:PAUSE 2
680   NEXT:SPRITE 1:SPRITE Ø,
XA*8+12,YA*8+19
```

リスト 30

マリオがカニさんをつかまえたらどうなりましたか？　なんと，カニさんが爆発(ばくはつ)してしまいました。

このように，動いているものどうしの判定(はんてい)というのは，その2つのものが，画面の同じ場所にあるかどうかを調べればいいのです。あとからいろいろ追加した，“回数を数える”，“爆発する”，などは2つのものが同じ場所にある，と判定されたあとにつけ加えるものです。

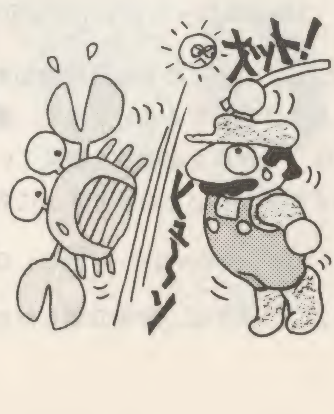
●3つ動かしてみる

それでは，もう1つ例としてプログラムを紹介(しょうかい)しましょう。

リスト 31

※BG GRAPHICモードで四方の壁を作ってからプログラムをRUNさせてください。

```
100 CLS:CGSET 1,1:VIEW
110 SPRITE ON
120 PA=Ø:XA=13:YA=1Ø:FM=Ø:F=
1
130 PD=192:XD=13:YD=5:DX=1
    *200〜320行までは，リスト29と同じものを打ち込む
400 '-----
410 IF FM<>1 THEN PAUSE 1:GO
TO 6ØØ
420  NB=NB-1:KB=ASC(SCR$(XB+
BX,YB+BY))
430   IF NB>=Ø AND KB=32 AND
(XB<>XD OR YB<>YD) GOTO 49Ø
440    FM=Ø:SPRITE 1
450    IF XB<>XD OR YB<>YD GO
TO 6ØØ
460    BEEP:SC=SC+1Ø:IF BY<>Ø
 THEN SC=SC+1Ø
470    LOCATE 13,21:PRINT SC
480    GOTO 6ØØ
490   XB=XB+BX:YB=YB+BY
500   SPRITE 1,XB*8+16,YB*8+2
3
510   GOTO 7ØØ
600 KG=STRIG(Ø):IF KG<>8 GOT
O 7ØØ
610   NB=1Ø:FM=1:XB=XA:YB=YA
620   IF KK<4 GOTO 65Ø
630    IF F=1 THEN XB=XB+1
```



```
640    IF F=2 THEN XB=XB-1
650   IF KK=1 THEN BX=1 :BY=0

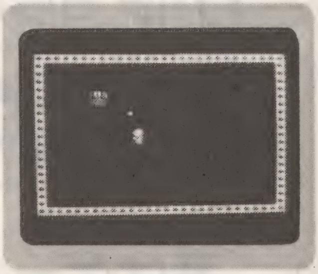
660   IF KK=2 THEN BX=-1:BY=0

670   IF KK=4 THEN BX=0 :BY=1

680   IF KK=8 THEN BX=0 :BY=-
1
690   DEF SPRITE 1,(2,0,0,0,0
)=CHR$(223)
700 '-----
710 K$=SCR$(XD+DX,YD)
720 IF K$<>" " THEN DX=-DX
730 XD=XD+DX
740 DEF SPRITE 3,(3,1,0,0,0)
=CHR$(PD)+CHR$(PD+1)+CHR$(PD
+2)+CHR$(PD+3)
750 SPRITE 3,XD*8+16,YD*8+20

760 PD=388-PD
800 '-----
810 GOTO 200
```

写真 14



これも，BG GRAPHICで四方の壁を作ってからRUNしてください。すると，カニさんが左右に動きますから，マリオを動かしてカニさんにボールをぶつけてください。今度は，カニさん，マリオ，ボール，と３つのものが出てきます。さっきやった「マリオがカニさんをつかまえる」プログラムでは，マリオとカニさんの２つしか出てきませんでしたが，判定のしかたはほとんど変わりません。今度は，「ボールがカニさんに当たったかどうか」だけを判定すればよいのですから，やっぱり動くもの２つの判定ということになるのです。マリオについては，なんの判定もされません。ですから，一見むずかしそうに見えるプログラムでも，基本はよく似たものが多いのです。

それから，このプログラムでは，得点にちょっとくふうがしてあります。ボールを横方向（右か左）に投げてカニさんに当たったときは10点，縦方向（上か下）に投げて当たったときは20点，というように，よりむずかしい方が点が高くなっています。これも，ゲームによくあるパターンですね。

最後にもうひとこと。このプログラムは，完成したゲームというわけではありませんから，いろいろやってみると「おかしいな？」と思うところもあるでしょう。たとえば，マリオをカニさんに合わせて動かしながら [A] ボタンを押し続けると，どんどん得点が入ってしまうことなどです。ですから，本当にゲームとして完成させるときには，マリオの動ける範囲を限定したり，カニさんの動き方をもっと複雑にしたり，障害物を置いたり，いろいろなくふうをするのです。

第5章 効果テキメン編



1. 音はゲームに欠かせない

ゲームに音は欠かせないものです。「マリオブラザース」,「ドンキーコング」,「ピンボール」などを頭に思い浮かべてみてください。いろいろな音が効果音として入っていて，ゲームのおもしろさをより盛り上げていますね。音を聞いて，どんなゲームだったかを思い出す人も多いと思います。それほど，音とゲームはお似合いなのです。

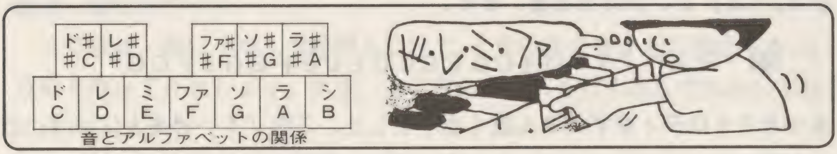
これまで画面の動きについて説明してきたわけですが，音といえばBEEPという，きまった音を出す命令しか使いませんでした。これではゲームとしてはつまらないですね。ですから，ここらで音についていろいろやってみましょう。

●音を出すにはPLAY命令

ファミリーベーシックには，PLAY 命令という，音を出す命令があります。くふうしだいでいろいろな音を出すことができますが，まず簡単なものからやってみましょう。

```
100  PLAY "C"
```

まず，このプログラムを打ち込んでRUNしましょう。音がしました。これが「ド」の音です。PLAY 命令で音を出すときは，このように「C」とか「D」とかアルファベットで音を指示するようになっています。



音とアルファベットの関係

図 14

そして，このアルファベットを並べておくと，その順に音を出していきます。たとえば，

```
100 PLAY "CEGEC"
```

なら，ドミソドミです。

●休符はR

では「C」と「E」の間に「R」を入れて，

```
100 PLAY "CREGEC"
```

にしてRUNしてください。ドとミの間に音の出ないときがありましたね。つまり「R」はお休みなのです。次は，

```
100 PLAY "CDECCDECEFGGEFGG"
```

を打ち込んでRUNしてみてください。どこかで聞いたメロディですね。

●音程はオクターブの「O」

では今度は，最初の「C」の前に「O 1」(オー・イチ) を入れて，

```
100 PLAY "O 1CDECCDECEFGGEFGG"
```

にしてからもう1度RUNしてみましょう。音が全体に低くなりました。「O」(オー)は，オクターブを決める命令で，「O」(オー)のあとにつける数字は，「0」(ゼロ)から「5」までです。つまり6オクターブ(ド～シまで6組分)の高さの音を出すことができるのです。「O」(オー)の後につける数字をいろいろ変えてRUNしてみてください。ずいぶん低い音から高い音まで出ますね。もとの高さに戻したいときは「O 3」(オー・サン)です。

●テンポをつける

次は，音を出すテンポを変えてみましょう。テンポは，音を出す速さのこ

とで，「T」という命令を使います。

100　PLAY "T1O1CDECCDECEFGGEFGG"

さっきのメロディがずいぶん速くなりました。この「T」のあとに「1」という数字がついていますが，この「1」が一番速く，「8」にするといちばん遅くなります。ためしに「8」に直してRUNしてみてください。じれったいでしょう。もとのテンポに戻したいときは「T4」です。

ではこの辺で，低い方から高い方までどのくらいの音が出るか聞いてみましょう。

```
100 A$="CDEFGAB"
110 PLAY "T1"
120 FOR I=0 TO 5
130  FOR J=1 TO 7
140   P$="O"+RIGHT$(STR$(I),
1)+MID$(A$,J,1)
150   PLAY P$
160  NEXT
170 NEXT
180 PLAY "O3T4"
```

リスト 32

これが音の高さの全域ですが，「#」をつけて半音を出すこともできます。

100　PLAY "CDE"

はドレミですが，ドとレ，レとミの間に半音を入れてみると，

100　PLAY "C#CD#DE"

となるわけです。これで，音の高さをすべて出すことができますね。

今やってみたオクターブやテンポは，1度命令しておくと，そのあとずっと有効になります。ですからたとえば，

100　PLAY "T2O4"

のように，最初に1回命令しておけばいいのです。その後で「オクターブを変えたい」などの必要が出てきたら，そのつど「O」(オー)「T」を使って命令することになります。

●音に変化をつける。

さて次は，音に1つ1つ数字をつけて，「その音の長さだけ指定する」というのをやってみましょう。まず，次のプログラムを打ち込んでRUNしてみてください。

```
100  PLAY  "T 2 O 2"
110  PLAY  "C 5 D E F G F E D C"
```

ドレミファソファミレドと，音が上がって下がりますね。では，110行の方を次のように直してみてください。

```
100  PLAY  "T 2 O 2"
110  PLAY  "C 5 D E F G F [2] E D C"
```

今度は，音が上がっていった後，かけ足で下がってきます。最初の「C」のあとに「5」がついていますから，ドレミファソまでは「5」の長さの音，つまり，4分音符の長さで音を出していきます。そして2番目の「F」のところまで進むと「2」という数字がついていますから，それからあとの音は符点16分音符の長さで音を出していく，となるわけです。つける数字と音の長さの関係は，次の図15のようになっています。

音の長さ	付ける数字	4分音符を1とした時の音の長さ	32分音符を1とした時の音の長さ
32分音符	0	8分の1	——
16分音符	1	4分の1	2倍
付点16分音符	2	8分の3	3倍
8分音符	3	半分	4倍
付点 8分音符	4	4分の3	6倍
4分音符	5	——	8倍
付点 4分音符	6	1.5倍	12倍
2分音符	7	2倍	16倍
付点 2分音符	8	3倍	24倍
全音符	9	4倍	32倍

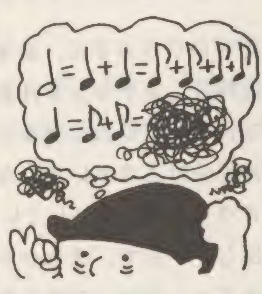


図 15

今やってみた例の場合，ドレミファソを4分音符で，その後のファミレド

を4分音符の3/8の長さの音で出したのですから，ゆっくり上がってかけ足で下がったわけです。

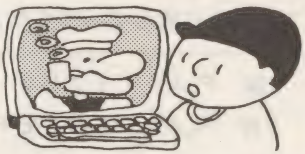
音の長さをもとに戻したいときは，「5」の数字をつけたお休みを利用して，

```
100 PLAY "R 5"
```

としましょう。

では，今までやったことでどんなことができるか，1つやってみましょう。

```
100 PLAY "T2O2"
110 PLAY "G3GGFREG8"
120 PLAY "A3FAO3C5O2A3G8"
130 PLAY "A3FAO3CO2BA"
140 PLAY "GAGERG"
150 PLAY "GGGARBO3C8"
160 PLAY "O3T4R5"
```



リスト 33

有名なあのメロディが聞こえてきました。160行は，音の長さや高さをもとに戻すために入っているものです。あとで混乱しないために，こういうクセをつけておく方がいいでしょう。

●3重和音だってできる。

これまでは音が1つしか出ていませんでしたが，PLAY命令では，音を重ねることができます。

```
100 PLAY "T2O1:T2O2"
110 PLAY "C3DEDC:E3FGFE"
```

このプログラムでは，音を2つ重ねています。1つの音を出すのをチャンネルと呼び，3チャンネルありますから3重和音まで出すことができます。そして，プログラムを見ると「：」(コロン)がありますが，この「：」がチャンネルを分けるさかいめになっています。このプログラムでは，チャンネルAでオクターブ1のドレミレド，チャンネルBでオクターブ2のミファソファミと出しています。それから，テンポやオクターブなども，それぞれのチャンネルごとに命令してやらなければなりません。たとえば，

100 PLAY "T2O1:T**4**O2"

```
110 PLAY "C3DEDC:E3FGFE"
```

のようにすると，チャンネルAのテンポは速く，Bのテンポは遅いですから，同時に終わってくれません。

ではさっきのリスト33を，A・Bチャンネルを使ってやってみましょう。

```
100 PLAY "T2O2:T2O2"
110 PLAY "G3GGFREG8:E3EEDRCE
8"
120 PLAY "A3FAO3C5O2A3G8:F3D
FA5F3E8"
130 PLAY "A3FAO3CO2BA:F3DFAG
F"
140 PLAY "GAGERG:EFECRE"
150 PLAY "GGGARB:EEEFRG"
160 PLAY "O3C1O2BO3CO2B:E1FE
F"
170 PLAY "O3CO2BO3C4:EFE4"
180 PLAY "O3T4R5:O3T4R5"
```

リスト 34

さっきより，いっそうよくなったでしょう。

●音質を変える。

音の質とは，音色や音の余いんのことです。

まず，次のプログラムを打ち込んでRUNしてみてください。

```
100 PLAY "O1T2M0V15Y0"
110 PLAY "D5R3D5D3D5R3F5E3E5
         D3D5#C3D6"
```

これは，「ヤラレタッ!!」というときによく使うメロディですね。この音は，普通の音で出しています。では，100行を次のように直して，RUNしてみてください。

```
100 PLAY "O1T2M0V[3]Y0"
```

まんなかあたりにある，「M」の後の数字が「0」(ゼロ)のときに，「V」の後の数字を変えると，音の大きさを指定することができるのです。数字は 0〜15までで，0がいちばん小さく，15がいちばん大きい音になっています。で

は，次はVを15に戻し，Mを1にしてください。

```
100  PLAY "O1T2M1V15YØ"
```

何だかもとどおりの音に戻ったように聞こえます。
では，Mは1のままでVを3にしてください。

```
100  PLAY "O1T2M1V3YØ"
```

今度は，はっきり違いますね。はねるような音になりました。「M」というのはエンベロープといって，1つの音をだんだん小さくする命令なのです。「M」の後の数字で,「M」命令を有効にする(1)か無効にする(Ø)かを決めるのです。そして，「M1」のときには，「V」は音の大きさではなく，どのくらいの速さで音を小さくするか，その程度を決める命令になります。ですから「M1V3」としたときには，速く小さくなるために，はねるような音になったわけです。では，もうちょっと感じを出してみましょう。

```
100  PLAY "O1T4M1V9YØ"
110  PLAY "D6D5D3D6F5E3E5D3D5
#C3D6"
```

だいぶ，「ヤラレタ」感じになりました。では，最後に「Y」です。100行のいちばんうしろの方にありますね。「Y」の後には，Ø～3までの数字を指示することができ，この数字によって4種類の音色にできます。「YØ」のØを1，2，3に変えてRUNしてみてください。

●音色プログラムを1つ

では，まとめに音のプログラムを紹介しましょう。
このプログラムは，100行と110行を直すと，いろいろな音が出ますからやってみてください。たとえば，

```
100  A$="CGCGCG"
110  B$="345345345"
```

や，

```
100  A$="CGCGCG"
```

```
110  B$="Ø1Ø1Ø1Ø1Ø1"
```

```
100  A$="EBDACG"
110  B$="4545454545"
120  L=LEN(B$)
130  PLAY "T1MØV15YØRØ"
140  '-----
150  FOR I=1 TO L
160    PLAY "O"+MID$(B$,I,1)+A
$
170  NEXT
```

リスト 35

などです。ただし，100行の〝 〃の中はA～Gのアルファベットと＃記号，110行の〝 〃の中はØ～5(オクターブ) の数字だけ使うようにしてください。文字の数はあまり多くなければ，適当でかまいません。

ではもう1つ。

```
100  A$="Ø12345"
110  B$="CDEFGAB"
120  PLAY "T1MØV15YØR2"
130  '-----
140  FOR I=Ø TO 49
150    P$="O"+MID$(A$,RND(6)+1
;1)+MID$(B$,RND(7)+1,1)
160    PLAY P$
170  NEXT
```

リスト 36

このプログラムは，6オクターブ分の音から乱数で1つを選び，50回音を出すものです。100行の〝 〃の中を5ばかりにすると，かわいらしい音になり，Øばかりにするとおそろしげな音になります。

2. 色のセンスがゲームをきめる

●PALET命令

これまで，マリオやカニさんやペンペンをいろいろ画面に出してきました

ね。そして，その色については，CGSET命令とDEF MOVEやDEF SPRITEの配色コードを変えて，ちょっと違う色も試してみました。でも，その方法だと12通りの色の組み合わせから選ぶだけでした。ところが，じつはPALET命令というものを使うと，これを自由に変えることができるのです。では，実験してみましょう。次のプログラムを打ち込んでRUNしてみてください。

```
100 CLS:CGSET 1,1
110 SPRITE ON
120 P=0
130 '-----
140 DEF SPRITE 0,(0,1,0,0,0)
=CHR$(P)+CHR$(P+1)+CHR$(P+2)
+CHR$(P+3)
150 SPRITE 0,120,120
200 '-----
210 LOCATE 3,1:INPUT "カオノ イロ
    -- ";A$:LOCATE 17,1:PRI
NT " "
220 LOCATE 3,2:INPUT "フクトボウシ
ノ イロ -- ";B$:LOCATE 17,2:PRI
NT " "
230 LOCATE 3,3:INPUT "ズボンノ イ
ロ    -- ";C$:LOCATE 17,3:PRI
NT " "
240 A=VAL(A$):B=VAL(B$):C=VA
L(C$)
250 PALET S 0,14,A,B,C
260 GOTO 200
```

リスト 37

RUNすると，見なれたマリオ(WALK 1)が出てきて，顔の色を何色にするか聞いてきますから，図16の表にある中から適当な数字(0～60)を選んでキーインし，RETURN キーを押してください。次は，服と帽子の色，ズボンの色も同じように数字をキーインし，RETURN キーを押してください。すると，マリオの色があなたのきめた色に変わりましたね。どんな色になってしまったのか，ここからではわかりませんが……。もとの色に戻したかったら，48,22,1をそれぞれキーインしてください。プログラムをやめたいときは，STOP キーです。

さて，このように，どのキャラクタでも，パレット命令で自由に色を変えることができるのです。色は図16のようにたくさんありますから，かなりい

〈52色の色コード〉

色相	10 進	10 進	10 進	10 進	
	0	16	32	48	灰色〜白色
青系	1	17	33	49	
青系	2	18	34	50	
青系	3	19	35	51	
青系	4	20	36	52	
赤系	5	21	37	53	
赤系	6	22	38	54	有色
赤系	7	23	39	55	
赤系	8	24	40	56	
緑系	9	25	41	57	
緑系	10	26	42	58	
緑系	11	27	43	59	
緑系	12	28	44	60	
	13	29	45		
	14	30	46		クロ
	15	31	47		

暗←――――→明



図 16

ろいろなバリエーションにできます。

●PALET S命令

さて，色を変える命令はPALET S 命令ですが，このプログラムでは 250 行で使っています。この命令は，ちょっとむずかしいので説明しておきましょう。

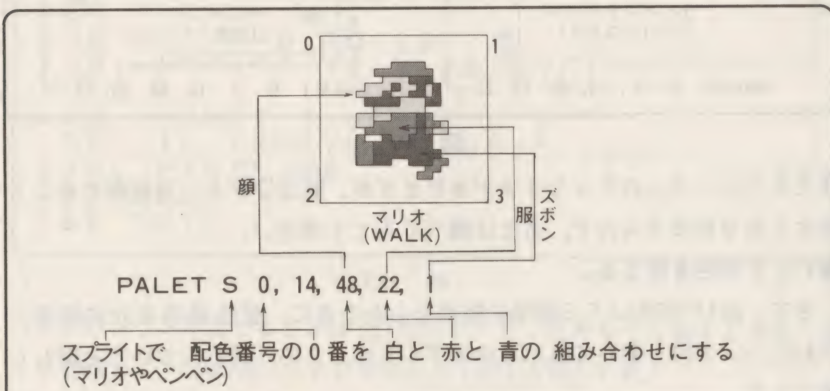


図 17

マリオはよーく見ると，白(48)，赤(22)，青(1)の3つの色でできています。そして，PALET S 命令では，終わりの3つの数字でそれぞれの色をき

めることができます。「配色番号の0番」をということについては，DEF SPRITE命令に出てきた配色番号を思い出してください。このプログラムにも140行にありますが，マリオの色の組み合わせをきめるときに，CGSETとDEF SPRITEの2つできめました。そしてDEF SPRITEには，配色番号という項目があって，そこを0にしましたね。PALET S命令の配色番号は，これを指すものです。ですから，DEF SPRITE命令で配色番号「3」を使っているキャラクタの色を変えたいときは，SPRITE S命令でも配色番号を「3」にしなければなりません。

さてマリオでは，PALET S命令の終わりの3つの数字が，それぞれ顔，服，ズボンの色に当てはまりますが(図17)，ほかのキャラクタではどう当てはまるのかは，次のようになっています。

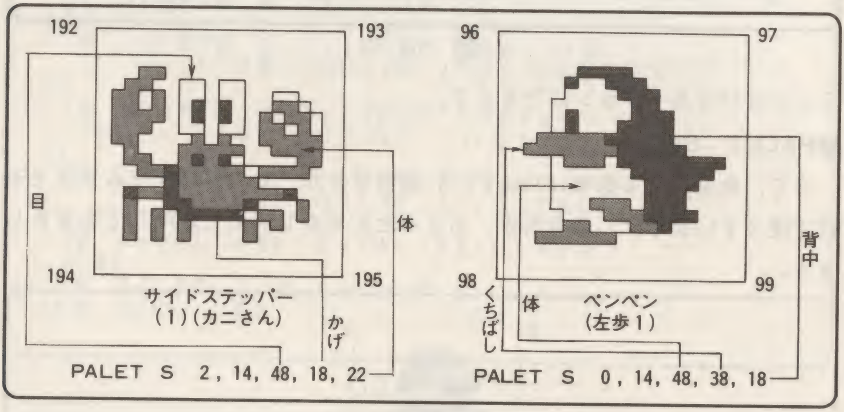


図 18

まだまだたくさんのキャラクタがありますが，残念ながら，全部載せることはとてもできませんので，あとは調べてみてください。

●バックの色を変える。

さて，図17でPALET S命令の説明をしたときに，配色番号の次の数字，「14」について何も書いてありませんでしたね。この「14」についてお話ししましょう。

今みなさんが見ている画面は，何も描かれていない部分は全部黒ですね。つまり黒板のようになっているのですが，この「14」という数字を変えると，画面全体のバックの色を変えることができるのです。実験してみましょう。

リスト37のプログラムの250行を次のように直してRUN し，さっきと同じように数字を３つ入力してみてください。

250 PALET S Ø, 22, 48, 22, 1

３回目の RETURN を押したとたん，画面がまっ赤になりましたね。この「22」という色を図16の表で見てみると，あたりまえですが，赤です。

では，バックの色をもとに戻しておきましょう。STOP C G S E T 1 RETURN と押してください。そうなのです。CGSET命令を実行すると，PALET S 命令で指定してあった色も，自動的にもとに戻るのです。

それから，バックの色を変えようとするときは，配色番号の数字を「Ø」にしておいてください。「Ø」以外の数字になっていると，いくら「14」のところを変えても，バックの色は変わりません。

では，バックの色をØ～6Øまで変えてみるプログラムを載せておきましょう。これは，ぜひ打ち込んでRUNしてみてください。これまでのパソコンでは表現できなかったような中間色が，とてもキレイです。

```
100 CLS:CGSET 1,1
110 LOCATE 9,1:PRINT "ハイショクバ
ンゴウ"
120 '-----
130 FOR I=Ø TO 6Ø
140  LOCATE 12,3:PRINT I;"  "
150  PALET S Ø,I,Ø,Ø,Ø
160  IF INKEY$="" GOTO 16Ø
170  PAUSE 2Ø
180 NEXT
190 GOTO 12Ø
```

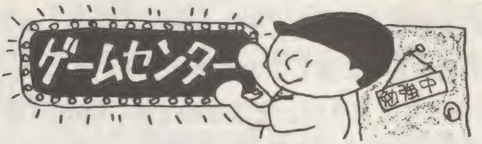
リスト 38

このプログラムを止めるときは STOP キーで，色をもとに戻すときは，さっきと同じようにCGSET 1 と打ち込んで RETURN です。

長々といろいろやってきました。どうですか？みんな覚えられましたか？

このあとは，楽しいゲームが４本です。なるべくプログラムを見やすく，わかりやすく，作ってありますから，これからBASICを勉強しようとしている人は，参考にしてください。

第6章 ゲームライブラリー



この章には，４本のゲームプログラムが載っています。「スピナーパニック」以外は，BG GRAPHICモードでバックの絵を作ってから，プログラムをRUNするようにしてください。BG GRAPHICは表になっています。たとえば「D4Ø」となっていたら，キャラクタテーブルのＤ列４番目のキャラクタを色Ø（青）で置くという意味です（付録２参照）。

ではゲームを楽しんでください。

1. スピナーパニック

このゲームには，BG GRAPHICはありません。プログラムだけで動きます。

プログラムをRUNさせると，青い枠が出てきます。あなたはカニさんで，この枠の中を自由に，８方向に動きまわることができます。カニさんは，コントローラ１で動かしてください。

さて，カニさんが追いかけるのはスピナーです。スピナーは次々と８個出てきますが，カニさんが捕まえることのできるホンモノのスピナーは，たった１個だけで，あとは全部ニセモノです。カニさんは，ワラワラと動きまわる８個のスピナーの中から，ホンモノを見つけ出し，追いかけてつかまえなければなりません。うまくつかまえることができると，スピナーは爆発し，また新しいホンモノのスピナーが枠の中のどこかにあらわれて動き出します。スピナーを１回つかまえると100点で，制限時間以内にできるだけたくさん

つかまえてください。1500点以上出るようになれば，まあまあです。

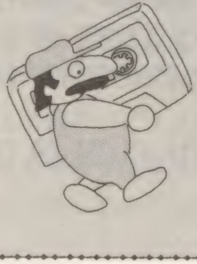
さて，慣れないとむずかしいでしょうから，ちょっとだけ，ホンモノのスピナーを見分けるヒントを出しておきましょう。カニさんが止まっているときには，ホンモノはニセモノよりも少し速く動き，カニさんが動いているときには，逆に少しゆっくり動きます。

スピナーパニック・プログラム

```
10 'ｽﾋﾟﾅｰ ﾊﾟﾆｯｸ
100 CLEAR:CLS:CGSET1,2:SPRIT
EON:PLAY"T100V15R0":PL$="CDE
FGAB"
110 FOR I=1 TO 7:POSITION I,
 120,100:NEXT
120 FOR I=0 TO 27 STEP 27:FO
R J=1 TO 22:LOCATE I,J:PRINT
 CHR$(193);:NEXT:NEXT
130 FOR I=1 TO 26:FOR J=0 TO
 23 STEP 23:LOCATE I,J:PRINT
 CHR$(193);:NEXT:NEXT
140 PX=120:PY=100:PP=192:GOS
UB1000:SC=0:PE=144:GOSUB2000
:NM=0
150 LOCATE 7,0:PRINT" SCORE
     0 "
200 R=RND(10):IF R<>0 GOTO30
0
210 RA=1+RND(7):RB=1+RND(8)
220 X=XPOS(RA):Y=YPOS(RA)
230 DEF MOVE(RA)=SPRITE (7,R
B,2,255,0,0)
240 POSITION RA,X,Y:MOVE RA
300 K=STICK(0):IF K=0 GOTO
 400
310 IF(K=9 OR K=1 OR K=5) AN
DPX<216 THEN PX=PX+4
320 IF(K=6 OR K=2 OR K=10)AN
DPX>24  THEN PX=PX-4
330 IF K>3 ANDK<7  ANDPY<192
 THEN PY=PY+4
340 IF K>7 ANDK<11 ANDPY>32
 THEN PY=PY-4
350 GOSUB1000
400 IF FE=1 GOTO 500
410 FE=1:ME=2*(10+RND(30)):D
E=1+RND(8):AX=0:AY=0
420 IF DE=1 THEN AY=-2
430 IF DE=3 THEN AX=2
440 IF DE=5 THEN AY=2
450 IF DE=7 THEN AX=-2
460 IF DE=2 THEN AX=2 :AY=-2
470 IF DE=4 THEN AX=2 :AY=2
480 IF DE=6 THEN AX=-2:AY=-2
490 IF DE=8 THEN AX=-2:AY=-2
500 EX=EX+AX:IF EX>255 OR EX
<0 THEN EX=EX+256*SGN(128-EX
)
510 EY=EY+AY:IF EY>255 OR EY
```

スピナーパニック・プログラム（続き）

```
<0 THEN EY=EY+256*SGN(128-EY
)
520 GOSUB1100
530 ME=ME-2:IF ME<0 THEN FE=
0
600 IF ABS(PX-EX)>4 OR ABS(P
Y-EY)>4 GOTO 700
610 PB=176
620 FOR I=0 TO 9
630 DEF SPRITE0,(0,1,0,0,0)=
CHR$(PB)+CHR$(PB+1)+CHR$(PB+
2)+CHR$(PB+3)
640 SPRITE0,PX,PY:PB=356-PB:
PLAY MID$(PL$,RND(7)+1,1)
650 NEXT
660 PLAY"O5CDEDCO0":SPRITE0:
FE=0:GOSUB2000
670 SC=SC+100:LOCATE15,0:PRI
NT RIGHT$("    "+STR$(SC),5)
680 GOTO200
700 NM=NM+1:IF NM<2000 GOTO
 200
710 CUT 1,2,3,4,5,6,7
720 S$="-GAME--OVER-"
730 FOR I=1 TO LEN(S$)
740 LOCATE8+I,23:PRINT MID$(
S$,I,1);:READ P$:PLAY"T1"+P$
750 NEXT
760 DATA O4B5,A,E,B,A,E,G,A,
B,A,E,E
770 LOCATE0:0:END
1000 DEF SPRITE2,(3,1,0,0,0)
=CHR$(PP)+CHR$(PP+1)+CHR$(PP
+2)+CHR$(PP+3)
1010 SPRITE2,PX,PY:PP=388-PP
:RETURN
1100 DEF SPRITE1,(0,1,0,0,0)
=CHR$(PE)+CHR$(PE+1)+CHR$(PE
+2)+CHR$(PE+3)
1110 SPRITE1,EX,EY:PE=292-PE
:RETURN
2000 EX=4*(7+RND(50)):EY=4*(
7+RND(40)):RETURN
```



★BASIC・V2.0・テープサービスのご案内

本書に掲載したプログラムのテープサービスを行なっております。キー入力の手間を省いて，プログラムの勉強をスピーディに進めるためにお役立てください。ご注文は(株)マイクロハウス(宛先は巻末奥付参照)宛，1800円(送料共)を現金書留にてお送りください。

（注：ＢＡＳＩＣ・Ｖ1.0のテープサービスは行なっておりません。）

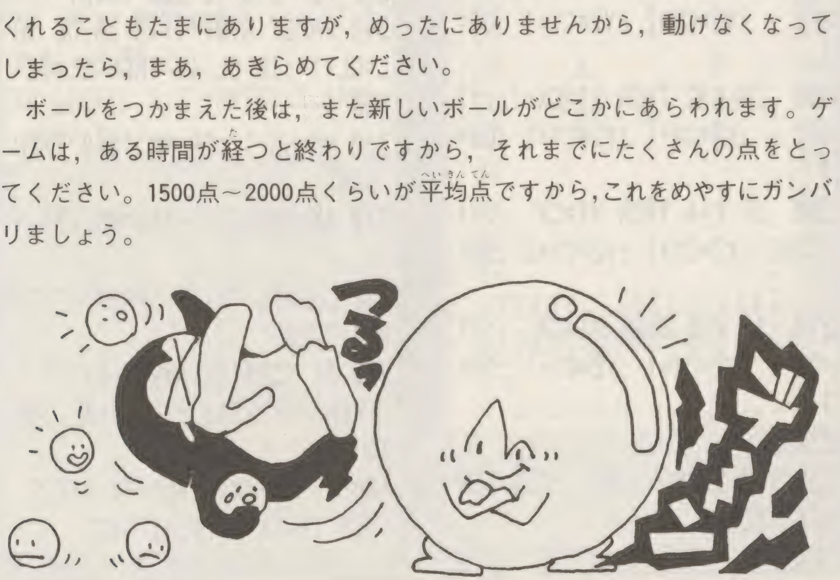
2. ペンペン大追跡

このゲームは，表にしたがってBG GRAPHICを描いてから，プログラムを動かすようにしてください。

プログラムをRUNすると，BG GRAPHICで描いたフィールドが出てきます。主人公はペンペンで，コントローラ1の上下左右どれかのキーを押すと出てきます。ペンペンが動けるのは，フィールドの中の何もない所だけで，方向は上下左右の4方向です。

さて，フィールドの中では不思議なボールが動きまわります。ペンペンがこのボールを追いつめてつかまえると得点になります。得点は，ボールをつかまえたときのフィールドの色によって，10点から120点までいろいろになります。どの色のときに120点になるかは，皆さん自分で見つけてください。また，つかまえたボールはつかまえた位置に障害物となって残ってしまいます。ですから，どこでつかまえ，つかまえた後ペンペンがどちらへ進むか，などをよく考えてやらないと，ペンペンが動けなくなってしまうこともありますから，注意してください。動きまわっているボールが，障害物を消してくれることもたまにありますが，めったにありませんから，動けなくなってしまったら，まあ，あきらめてください。

ボールをつかまえた後は，また新しいボールがどこかにあらわれます。ゲームは，ある時間が経つと終わりですから，それまでにたくさんの点をとってください。1500点～2000点くらいが平均点ですから,これをめやすにガンバリましょう。



ペンペン大追跡・プログラム

```
10 'ﾍﾟﾝﾍﾟﾝ ﾀﾞｲﾂｲｾｷ
100 CLS :VIEW :SPRITE ON :CG
SET 1,2 :CGEN 2:PLAY "T1R0"
110 CX=13 :GX=(CX+2)*8 :CY=2
 :GY=(CY+3)*8-1 :SC=0 :NM=0
120 LOCATE 8,0 :PRINT "SCORE
 □□□□□"
130 LOCATE 15,0 :PRINT RIGHT
$("     "+STR$(SC),5)
140 CL=33 :TX=2+RND(24) :TY=
2+RND(16)
150 IF SCR$(TX,TY)<>" " THEN
 140
160 '□□□□□
170 K=STICK(0) :IF K=0 THEN
 320
180  IF K=1 THEN X1=CX+2 :Y1
=CY   :X2=CX+2 :Y2=CY+1 :DR=
3
190  IF K=2 THEN X1=CX-1 :Y1
=CY   :X2=CX-1 :Y2=CY+1 :DR=
7
200  IF K=4 THEN X1=CX   :Y1
=CY+2 :X2=CX+1 :Y2=CY+2 :DR=
5
210  IF K=8 THEN X1=CX   :Y1
=CY-1 :X2=CX+1 :Y2=CY-1 :DR=
1
220 '---
230 K1$=SCR$(X1,Y1) :K2$=SCR
$(X2,Y2)
240 IF K1$<>" " OR K2$<>" "
THEN 320
250  DEF MOVE(0)=SPRITE(4,DR
,1,2,0)
260  POSITION 0,GX,GY :MOVE
0
270  IF K=1 THEN CX=CX+1 :GX
=GX+8
280  IF K=2 THEN CX=CX-1 :GX
=GX-8
290  IF K=4 THEN CY=CY+1 :GY
=GY+8
300  IF K=8 THEN CY=CY-1 :GY
=GY-8
310 GOTO 340
320 '-----
330 FOR I=0 TO 200 :NEXT
340 PALET B 0,14,48,33,CL :C
L=CL+1 :IF CL>44 THEN CL=33
350 '-----
360 NM=NM+1 :IF NM>1000 THEN
 510
370 AX=RND(3)-1 :AY=RND(3)-1

380 LOCATE TX,TY :PRINT " ":
PLAY"O2G"
390 IF SCR$(TX+AX,TY+AY)=" "
 THEN TX=TX+AX :TY=TY+AY :GO
TO 420
400 IF TX-AX>2 AND TX-AX<25
THEN TX=TX-AX:PLAY"O0C"
```

```
410 IF TY-AY>2 AND TY-AY<18
THEN TY=TY-AY:PLAY"O0E"
420 LOCATE TX,TY :PRINT CHR$
(207)
430 '-----
440 SH=0
450 IF TX-CX=0 OR TX-CX=1 TH
EN SH=SH+1
460 IF TY-CY=0 OR TY-CY=1 TH
EN SH=SH+1
470 IF SH<>2 THEN 160
480 PLAY "O4CEGFEDC"
490 SC=SC+(CL-32)*10
500 GOTO 130
510 '□□□□□
520 FOR I=0 TO 48 STEP 16 :P
ALET B 0,I,48,33,2 :NEXT
530  PLAY "B#AA#GG" :PALET B
 0,14,48,33,2
540 LOCATE 8,1 :PRINT " GAME
  OVER "
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0	F40	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F60
1	F40	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F60
2	F40	F60																									F40	F60
3	F40	F60																									F40	F60
4	F40	F60																									F40	F60
5	F40	F60						D40						F40	F60						D40						F40	F60
6	F40	F60					D40	D40	D40											D40	D40	D40					F40	F60
7	F40	F60						D40													D40						F40	F60
8	F40	F60												F40	F60												F40	F60
9	F40	F60																									F40	F60
10	F40	F60																									F40	F60
11	F40	F60																									F40	F60
12	F40	F60												F40	F60												F40	F60
13	F40	F60						D40													D40						F40	F60
14	F40	F60					D40	D40	D40											D40	D40	D40					F40	F60
15	F40	F60						D40						F40	F60						D40						F40	F60
16	F40	F60																									F40	F60
17	F40	F60																									F40	F60
18	F40	F60																									F40	F60
19	F40	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F60
20	F40	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F50	F60

3. マリオランド

このゲームも，表にしたがってBG GRAPHICを描いてから，プログラムを動かすようにしてください。

今度は，マリオがある島で活躍するゲームです。プログラムをRUNすると，BG GRAPHICで描いた島が出てきます。そして，コントローラ１の左右どちらかのキーを押すとマリオが出てきます。このマリオは，３つの動きができます。１つめは島の上を歩くことで，コントローラ１の←→キーを使います。２つめは↑↓キーを使って木に登るのですが，幹の部分までで葉の部分へは行けません。３つめはジャンプで，マリオが地面にいるときに適当な長さだけ↑キーを押します。この押した長さで高さがきまり，ジャンプしたい所でⒷボタンを押します。木に登っているときはジャンプはできません。またジャンプ中は左右に動けません。以上３つをうまく使い分けてください。

さて画面左から，いろいろな高さでトリさんが飛んできますから，あなたはマリオをいろいろ操作してⒶボタンでトリさんをぶってください。木に登ってぶつと10点，葉より下のトリさんをジャンプしてぶつと50点，葉の高さにいるトリさんをジャンプしてぶつと100点です。時間が来ればゲーム終了です。なお，島の端まで行くと海に落ちますが，減点にはなりません。

マリオランド・プログラム

```
10 'ﾏﾘｵﾗﾝﾄﾞ
100 CLS:CGSET1,0:SPRITEON:VI
EW:DIM BD(1,2):LOCATE11,9:PR
INT"SCORE"
110 FOR I=1 TO 2:J=196+4*I
120 DEFSPRITE I,(0,1,0,1,0)=
CHR$(J+1)+CHR$(J)+CHR$(J+3)+
CHR$(J+2)
130 NEXT:PLAY"T104V15R0"
140 MU=0:MX=120:MY=160:FJ=0:
MN=0:BB=1:BD(0,0)=1:BD(0,1)=  ↗
100:BD(0,2)=20+RND(100)
150 '
160 MN=MN+1:IF MN>1000 GOTO
 760
170 K1=STICK(0):IF FJ<>0 GOT
O480
180 IF K1=1 ANDMY=160 GOTO
 220
190 IF K1=2 ANDMY=160 GOTO
 240
200 IF K1=4 ORK1=8 GOTO 260
```

マリオランド・プログラム（続き）

```
210 GOTO480
220 'R
230 P=M:GOSUB1000:MX=MX+4:GO
TO370
240 'L
250 P=M:GOSUB1010:MX=MX-4:GO
TO370
260 'UD
270 IF MX<>60ANDMX<>180 GOTO
 340
280 MU=ABS(MU-1):P=5
290 IF MU=0 GOSUB1000
300 IF MU=1 GOSUB1010
310 IF K1=8 ANDMY>56  THEN M
Y=MY-4
320 IF K1=4 ANDMY<160 THEN M
Y=MY+4
330 GOTO370
340 'JC
350 IF FJ=0 ANDJC<160 THEN J
C=JC+4:PLAY"O5BO4"
360 GOTO570
370 '
380 SPRITE 0,MX,MY
390 M=M+1:IF M>2 THEN M=0
400 IF MX>=28 ANDMX<=212 GOT
O570
410 P=6:GOSUB1000
420 IF MX>120 THEN MX=MX+4
430 IF MX<120 THEN MX=MX-4
440 FOR I=MY TO 180
450  SPRITE 0,MX,I
460 NEXT:PLAY"O1F4EO4R0"
470 MX=120:MY=160:GOTO570
480 'J
490 K2=STRIG(0)
500 IF K2=4 ANDFJ=0 ANDJC>0
ANDMY=160 THEN FJ=1:JT=JC
510 IF FJ=0 THEN 570
520 IF JC>0 ANDFJ=1 THEN MY=
MY-4:JC=JC-4
530 IF JC<1 THEN FJ=2
540 IF FJ=2 ANDJC<JT THEN JC
=JC+4:MY=MY+4
550 IF JC=JT ANDFJ=2 THEN FJ
=0:JC=0
560 P=2:GOSUB1000:SPRITE 0,M
X,MY
570 'B
580 FOR I=0 TO 1
590 IF BD(I,0)<>0 GOTO610
600 BD(I,0)=1:BD(I,1)=0:BD(I
,2)=20+RND(100)
610 BD(I,1)=BD(I,1)+4:IF BD(
I,1)>240 THEN BD(I,0)=0:GOTO
 590
620 SPRITE BB,BD(I,1),BD(I,2
):BB=3-BB
630 NEXT:BB=3-BB
640 'P
650 IF STRIG(0)<>8 GOTO150
660 P=3:GOSUB1000:SPRITE 0,M
X,MY:PLAY"O2CO4"
670 FOR I=0 TO 1:S1=ABS(MX+4
```

マリオランド・プログラム（続き）

```
-BD(I,1)):S2=ABS(MY-4-BD(I,2
))
680 IF S1>4 OR S2>4 GOTO720
690 SC=SC+10:IF FJ<>0 THEN S
C=SC+40
700 IF BD(I,2)<48 THEN SC=SC
+50
710 PLAY"CDE":LOCATE11,11:PR
INT SC
720 NEXT
730 P=5:IF MY=160 THEN P=M
740 GOSUB1000:SPRITE 0,MX,MY
750 GOTO150
760 LOCATE 9,13:PRINT"GAME O
VER":END
1000 DEF SPRITE 0,(1,1,0,1,0
)=CHR$(4*P+1)+CHR$(4*P)+CHR$
(4*P+3)+CHR$(4*P+2):RETURN
1010 DEF SPRITE 0,(1,1,0,0,0
)=CHR$(4*P)+CHR$(4*P+1)+CHR$
(4*P+2)+CHR$(4*P+3):RETURN
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0							I33							
1						I03	I13	I23						
2					I03	I13	I13	I13	I23					
3				I03	I13	I13	I13	I13	I13	I23				
4							I43							
5							I43							
6							I43							
7							I43							
8							I43							
9							I43							
10							I43							
11							I43							
12							I43							
13							I43							
14							I43							
15							I43							
16							I43							
17							I43							
18							I43							
19			I03	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13
20	M60	M60	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43

	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0								I33						
1							I03	I13	I23					
2						I03	I13	I13	I13	I23				
3					I03	I13	I13	I13	I13	I13	I23			
4								I43						
5								I43						
6								I43						
7								I43						
8								I43						
9								I43						
10								I43						
11								I43						
12								I43						
13								I43						
14								I43						
15								I43						
16								I43						
17								I43						
18								I43						
19	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I23		
20	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	M60	M60

4. チェンジ・チェンジ

このゲームも，表にしたがってBG GRAPHICを描いてから，プログラムを動かすようにしてください。

これは，２人でやるゲームです。１人ではできません。プログラムをRUNすると，BG GRAPHICで描いたフィールドが出てきます。そして左右どちらかに「センコウ」（先攻）と出て，すぐにマリオが２人出てきます。先攻と出た方はマリオの色，もう１人は緑色をしています。

ここからゲームが始まります。先攻になった人は，すかさず緑のマリオを追いかけてぶってください。動けるのは上下左右４方向で，ぶつのは A ボタンです。残念ながら，マリオが緑色になってしまった方の人は，逃げまわってください。フィールドにいろいろあるものは目かくしですから，これをうまく使ってぶたれないようにしてください。

さて，ある時間が経つと音がして，マリオの色が入れ替わり，今度は今まで逃げていた人が追いかけ，追いかけていた人が逃げる番です。そして，またある時間が経つと音がして，色が入れ替わります。色の入れ替わりは全部で９回ですから，お互いに５回ずつ追いかけてぶつチャンスがあるわけです。

ゲームは，どちらかが20回相手をぶつとその人の勝ちとなります。また，制限時間以内に，どちらも20回打てなかったときは，判定となり，ぶった回数が多い方が勝ちになります。負けた方のマリオは，みじめな格好になってしまいますから，そうならないようにガンバリましょう。

チェンジ・チェンジ・プログラム

```
10 'ﾁｪﾝｼﾞﾁｪﾝｼﾞ
100 CLS:VIEW:CGSET1,1:SPRITE
ON:PLAY"T1V15O3R0"
110 F=RND(2):LOCATE3+18*F,10
:PLAY"GAB":PRINT"ｾﾝｺｳ"
120 FOR I=0 TO 3000:NEXT:LOC
ATE3+18*F,10:PLAY"AGF":PRINT
"    "
130 NM=0:CA=0:DA=1:XA=30:YA=
100:CB=2:DB=2:XB=210:YB=100:
IF F=1 THEN SWAP CA,CB
140 PA=4:GOSUB1000:SPRITE0,X
A,YA
150 PB=4:GOSUB1030:SPRITE1,X
B,YB
200 '
```

チェンジ・チェンジ・プログラム (続き)

```
210 NM=NM+1:IF(NM MOD200)=0
THEN PLAY"T1V15E0GEC":SWAP C
A,CB
220 IF NM>2000 GOTO800
300 KA=STICK(0):IF KA=0 THEN
 400
310 IF KA=1ANDXA<218 THEN XA
=XA+4:DA=1
320 IF KA=2ANDXA>24 THEN XA=
XA-4:DA=2
330 IF KA=4ANDYA<164 THEN YA
=YA+4
340 IF KA=8ANDYA>32 THEN YA=
YA-4
350 PA=PA+4:IF PA>8 THEN PA=
0
360 IF DA=1 GOSUB1000
370 IF DA=2 GOSUB1010
380 SPRITE0,XA,YA
400 KB=STICK(1):IF KB=0 GOTO
500
410 IF KB=1 ANDXB<218 THEN X
B=XB+4:DB=1
420 IF KB=2 ANDXB>24  THEN X
B=XB-4:DB=2
430 IF KB=4 ANDYB<164 THEN Y
B=YB+4
440 IF KB=8 ANDYB>32  THEN Y
B=YB-4
450 PB=PB+4:IF PB>8 THEN PB=
0
460 IF DB=1 GOSUB1020
470 IF DB=2 GOSUB1030
480 SPRITE1,XB,YB
500 NA=0:KA=STRIG(0):IF KA=0
 ORCA<>0 GOTO600
510 SWAP S,PA:NA=1
520 IF DA=1 THEN PA=12:GOSUB
1000
530 IF DA=2 THEN PA=12:GOSUB
1010
540 SPRITEO,XA,YA:SWAP S,PA:
FOR J=0 TO 20:NEXT
550 IF DA=1 GOSUB1000
560 IF DA=2 GOSUB1010
570 SPRITE0,XA,YA:PLAY"CD"
600 NB=0:KB=STRIG(1):IF KB=0
 ORCB<>0 GOTO700
610 SWAP S,PB:NB=1
620 IF DB=1 THEN PB=12:GOSUB
1020
630 IF DB=2 THEN PB=12:GOSUB
1030
640 SPRITE1,XB,YB:SWAP S,PB:
FOR J=0 TO 20:NEXT
650 IF DB=1 GOSUB1020
660 IF DB=2 GOSUB1030
670 SPRITE1,XB,YB:PLAY"BA"
700 IF ABS(XA-XB)>4 ORABS(YA
-YB)>4 THEN 200
710 IF NA=1 THEN SA=SA+1
720 IF NB=1 THEN SB=SB+1
730 PLAY"T1C":LOCATE3,21:PRI
NT SA:LOCATE20,21:PRINT SB
```

```
740 IF SA<20 ANDSB<20 GOTO20
0
800 XA=30:YA=100:CA=0:XB=210
:YB=100:CB=0:IF SA=SB THEN
 830
810 IF SB>SA THEN PA=24:CA=2
:GOSUB1000:GOTO 830
820 PB=24:CB=2:GOSUB1020
830 PLAY"E4GEC":SPRITE0,XA,Y
A:SPRITE1,XB,YB
840 LOCATE 9,10:PRINT"GAME
OVER":LOCATE0,0:END
1000 DEF SPRITE0,(CA,1,1,1,0
)=CHR$(PA+1)+CHR$(PA)+CHR$(P
A+3)+CHR$(PA+2):RETURN
1010 DEF SPRITE0,(CA,1,1,0,0
)=CHR$(PA)+CHR$(PA+1)+CHR$(P
A+2)+CHR$(PA+3):RETURN
1020 DEF SPRITE1,(CB,1,1,1,0
)=CHR$(PB+1)+CHR$(PB)+CHR$(P
B+3)+CHR$(PB+2):RETURN
1030 DEF SPRITE1,(CB,1,1,0,0
)=CHR$(PB)+CHR$(PB+1)+CHR$(P
B+2)+CHR$(PB+3):RETURN
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	
0	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	0
1	F33	F33	F33	F33									F33	F33	F33	F33									F33	F33	F33	F33	1
2	F33	F33	F33	F33									F33	F33	F33	F33									F33	F33	F33	F33	2
3	F33	F33	F33	F33		L31	L61	L61	L61	L71									L31	L61	L61	L61	L71		F33	F33	F33	F33	3
4	F33	F33				L41	M71	M71	M71	L41									L41	M72	M72	M72	L41				F33	F33	4
5	F33	F33				L41	M71	L31	L61	M01									L51	L61	L71	M72	L41				F33	F33	5
6	F33					L41	M71	L41					L31	L61	L61	L71					L41	M72	L41					F33	6
7	F33					L41	M71	L41					L41	M71	M72	L41					L41	M72	L41					F33	7
8	F33					L41	M71	L41			L31	L61	L41	M71	M72	L41	L61	L71			L41	M72	L41					F33	8
9	F33					L51	L61	M01			L41	M71	L41	M71	M72	L41	M72	L41			L51	L61	M01					F33	9
10	F33										L41	M71	L41	K01	K02	L41	M72	L41										F33	10
11	F33					L31	L61	L71			L41	M71	L41	M71	M72	L41	M72	L41			L31	L61	L71					F33	11
12	F33					L41	M71	L41			L51	L61	L41	M71	M72	L41	L61	M01			L41	M72	L41					F33	12
13	F33					L41	M71	L41					L41	M71	M72	L41					L41	M72	L41					F33	13
14	F33					L41	M71	L41					L51	L61	L61	M01					L41	M72	L41					F33	14
15	F33	F33				L41	M71	L51	L61	L71									L31	L61	M01	M72	L41				F33	F33	15
16	F33	F33				L41	M71	M71	M71	L41									L41	M72	M72	M72	L41				F33	F33	16
17	F33	F33	F33	F33		L51	L61	L61	L61	M01									L51	L61	L61	L61	M01		F33	F33	F33	F33	17
18	F33	F33	F33	F33									F33	F33	F33	F33									F33	F33	F33	F33	18
19	F33	F33	F33	F33									F33	F33	F33	F33									F33	F33	F33	F33	19
20	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	20
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	

●付録 1● キャラクタコード表

ファミリーベーシックのキャラクタコードを，一覧表で示します。ＢＡＳＩＣのCHR$（n），ASC（"A$"）による文字とコードの変換は，これに対応しています。キャラクタコード表Ａ，またはＢのキャラクタ，文字，記号は，バックグラウンド，スプライトの各面で使用できます。（どの面にどちらの表を使用するかの指定はＣＧＥＮ命令を使用します）

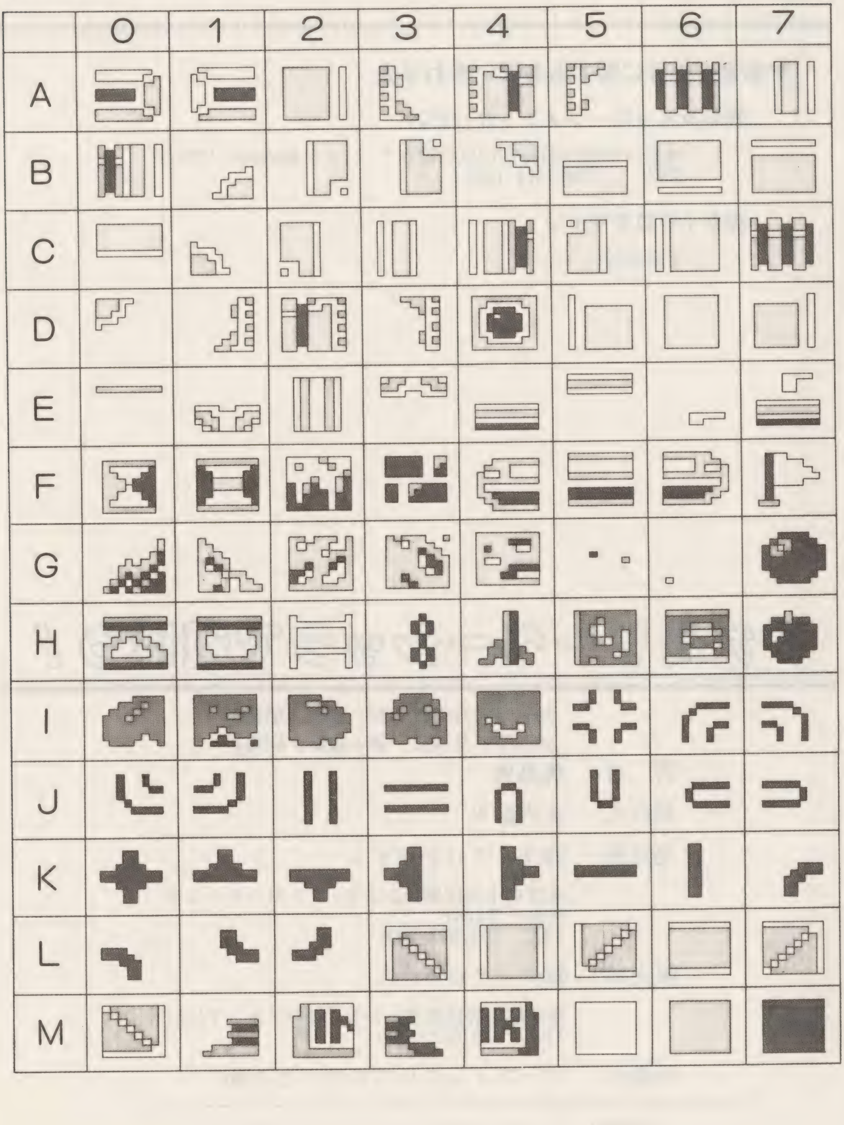
● キャラクタコード表Ａ（主に，スプライト用）

コード（10進数）は，キャラクタテーブルＡでアニメキャラクタの四隅に表記してある番号に対応します。

- アニメキャラクタを定義する場合は，このコード表にあるコード(10進数，または16進数)で指定してください。
- ０～255までの10進と16進の換算表としてもご利用ください。

コード（10進数）	説明	コード（10進数）	説明	コード（10進数）	説明	コード（10進数）	説明	コード（10進数）	説明	コード（10進数）	説明	コード（10進数）	説明
0	マリオ（WALK1）	40	レディ（JUMP）	80	アキレス（上１）	120	車（左１）	160	スターキラー（上）	200	ニットピッカー（１）	240	1
1		41		81		121		161		201		241	2
2		42		82		122		162		202		242	3
3		43		83		123		163		203		243	4
4	マリオ（WALK2）	44	レディ（スリップ 着地）	84	アキレス（上２）	124	車（左２）	164	スターシップ（左）	204	ニットピッカー（２）	244	COMPUTERの文字
5		45		85		125		165		205		245	
6		46		86		126		166		206		246	
7		47		87		127		167		207		247	
8	マリオ（WALK3）	48	レディ（はしご）	88	ニタニタ（１）	128	車（左上１）	168	スターシップ（左上）	208	レーザー（よこ）	248	
9		49		89		129		169		209		249	
10		50		90		130		170		210	レーザー（ななめ）	250	OPERATORの文字
11		51		91		131		171		211		251	
12	マリオ（JUMP）	52	レディ（DOWN）	92	ニタニタ（２）	132	車（左上２）	172	スターシップ（上）	212	レーザー（たて）	252	
13		53		93		133		173		213		253	
14		54		94		134		174		214	スプライト用ぬりつぶしの色(1),(2),(3)	254	
15		55		95		135		175		215		255	
16	マリオ（スリップ 着地）	56	ファイターフライ（１）	96	ペンペン（左歩１）	136	車（上１）	176	爆発（１）	216			
17		57		97		137		177		217	ミュージックボード用カーソル(1),(2),(3)		
18		58		98		138		178		218			
19		59		99		139		179		219			
20	マリオ（はしご）	60	ファイターフライ（２）	100	ペンペン（左歩２）	140	車（上２）	180	爆発（２）	2200	第１カーソル		
21		61		101		141		181		221	第２カーソル		
22		62		102		142		182		222	ランプ（１）		
23		63		103		143		183		223	ランプ（２）		
24	マリオ（DOWN）	64	アキレス（左１）	104	ペンペン（正面）	144	スピナー（１）	184	シェルクリーパー（１）	224	はねペン（１）		
25		65		105		145		185		225			
26		66		106		146		186		226			
27		67		107		147		187		227			
28	レディ（WALK1）	68	アキレス（左２）	108	ペンペン（後）	148	スピナー（２）	188	シェルクリーパー（２）	228			
29		69		109		149		189		229			
30		70		110		150		190		230			
31		71		111		151		191		231			
32	レディ（WALK2）	72	アキレス（左上１）	112	ファイアーボール（１）	152	スターキラー（左）	192	サイドステッパー（１）	232	はねペン（２）		
33		73		113		153		193		233			
34		74		114		154		194		234			
35		75		115		155		195		235			
36	レディ（WALK3）	76	アキレス（左上２）	116	ファイアーボール（２）	156	スターキラー（左上）	196	サイドステッパー（２）	236			
37		77		117		157		197		237			
38		78		118		158		198		238			
39		79		119		159		199		239			

●付録 2● キャラクタテーブル



▼本書の内容に関するお問い合わせ先

㈱O＆Kマネージメントサービス

神奈川県横浜市神奈川区台町7-2　ハイツ横浜604（〒221)
TEL　(045)314-1127

㈱マイクロデザイン

下記参照

ファミリーベーシックで遊ぼう！

1984年10月20日　初　版第１刷発行
1984年11月20日　第４版第１刷発行

著　者　鹿島博

発行人　武内静夫

発行所　㈱マイクロデザイン

東京都中央区築地3-2-3　シルバールームズ
1004（〒104)
TEL　(03)543-1601

販売元　㈱マイクロハウス

東京都中央区新富1-3-9　東光ビル（〒104)
TEL　(03)552-1530

印刷所　パシフィックリプロサービス㈱

MICRO DESIGN

￥980

Ⓓ

ファミリーベーシックで遊ぼう!

マイクロデザイン

『ファミリーベーシックで遊ぼう！』正誤表

このたびは、小社発行の『ファミリーベーシックで遊ぼう！』をお買い上げいただきまして、ありがとうございます。

本書中に、訂正がいくつかございますので、お知らせいたします。

【第４版】

訂正箇所	誤	正
●P37 リスト17の300行	SPRITE 0 (0,…	SPRITE 0,(0,…
●P45 リスト22の310行	…XA＊8＋12：…	…XA＊8＋12,…
●P60リスト31	行番号３３０を追加 ３３０　KK＝K　とする。	
●P69 リスト36の150行	…RND(６)＋１，１)…→，の上にヨゴレがあるため；と間違えやすい	
●P75 480行	…AY＝－２	…AY＝２

1985.1.10 現在

発売元
株式会社 マイクロハウス
〒104 東京都中央区新富1-3-9 東光ビル6F・7F
TEL 03(552)1530(代表)

発行元
株式会社 マイクロデザイン
〒104 東京都中央区築地3-2-3 シルバールームズ1004号
TEL 03(543)1601(代表)